NiceLabel Automation Guide d'Utilisation

Édition française

Rév. 15-06

© 2015 Euro Plus d.o.o. Tous droits réservés.

Euro Plus d.o.o.
Poslovna cona A 2
SI-4208 Šenčur, Slovenia
Tél.: +386 4 280 50 00
Fax: +386 4 233 11 48
www.nicelabel.com
info@nicelabel.com

Table Des Matières

Table des Matières **2**
Bienvenue dans NiceLabel Automation **5**
Conventions Typographiques **7**
Paramétrage de l'Application **8**
Architecture 8
Spécifications Système requises 8
Installation 9
Activation 10
Mode d'essai 11
Comprendre les Filtres **12**
Comprendre les Filtres 12
Configuration de Filtre de Texte Structuré 13
Configuration de Filtre de Données Non-structurées 18
Configuration de filtre XML 26
Paramétrage des noms d'Étiquette et d'Imprimante des données d'entrée 33
Configuration des Déclencheurs **34**
Comprendre les Déclencheurs 34
Définition des Déclencheurs 36
Utilisation de Variables 64
Utilisation des Actions 68
Test des déclencheurs 130
Protéger la Configuration du Déclencheur contre l'édition 132
Utilisation de la Couche de Transport Sécurisée (HTTPS) 133
Exécuter et Gérer les Déclencheurs **136**
Déployer la Configuration 136
Options de journalisation des évènements 137

Gestion des déclencheurs 137
Utilisation du journal d'évènements 139
Performances et Options de Feedback 141
Traitement Parallèle 141
Mise en cache de Fichiers 142
Traitement d'Erreurs 143
Mode d'impression Synchrone 144
Feedback de la tâche d'impression 145
Cluster haute disponibilité (Fail-over) 148
Cluster Répartition de Charge 148
Comprendre les Structures de Données 150
Comprendre les Structures de Données 150
Fichiers Binaires 150
Fichiers de Commande 151
CSV Composé 151
Données existantes 152
Base de données de Texte 152
Données XML 153
Référence et Résolution de Problèmes 155
Types de Fichiers de Commande 155
Commandes Personnalisées 162
Accès aux ressources réseau partagées 167
Accéder aux bases de données 168
Changer les paramètres par défaut d'Impressions Multiples 169
Compatibilité avec les produits NiceWatch 170
Contrôler le Service avec les paramètres de ligne de Commande 171
Introduction de Caractères Spéciaux (Codes de Contrôle) 173
Liste des Codes de Contrôle 174
Mode hors-ligne 175
Mode de Licence d'Imprimante 175
Fonctionnement en Mode de Service 175
Ordre de recherche pour les fichiers requis 177
Sécuriser l'accès à vos déclencheurs 178

Conseils et astuces d'utilisation de Variables dans les Actions 179
Mode de Traçage 179
Comprendre les paramètres d'imprimante et DEVMODE 180
Utiliser le même compte utilisateur pour configurer et exécuter les déclencheurs 182
Exemples **183**
Exemples 183
Support Technique **185**
Support en-ligne 185

Bienvenue Dans NiceLabel Automation

NiceLabel Automation est une application d'automatisation des actions répétitives. En général elle sera utilisée pour intégrer un processus d'impression d'étiquettes dans un système d'informations existant, telles qu'une application de bureau existante, des lignes de production et d'emballage, des systèmes de distribution et chaînes de distribution. Toutes les applications utilisées dans toutes les divisions et endroits de votre entreprise peuvent désormais imprimer des modèles d'étiquettes autorisés.

NiceLabel Automation représente le système d'entreprise idéal pour l'impression d'étiquettes en synchronisant les évènements de l'entreprise avec la production d'étiquettes. L'impression automatique sans interaction humaine est de loin la façon la plus effective d'éliminer les erreurs humaines et de maximaliser les performances.

Impression automatique d'étiquettes avec application à déclencheur basée sur 3 processus centraux.

Déclencheur

Les déclencheurs ont une fonction simple mais puissante qui vous aide à automatiser le travail. En fait, le déclencheur est une instruction de cause-à-effet : quand un évènement surveillé survient, effectue une action.

Il s'agit du traitement **IF .. THEN** (si...alors). Les déclencheurs sont utilises pour les choses qui se répètent.

L'impression automatique d'étiquettes est déclenchée par une opération dans l'entreprise. NiceLabel Automation est paramétré pour surveiller un dossier, un fichier, une communication ou un port. Quand une opération s'effectue dans l'entreprise, un changement de fichier ou des données entrantes sont détectés, ce qui active le processus d'impression d'étiquettes.

En savoir plus sur les différents Déclencheurs:

- Déclencheur de Fichier
- Déclencheur de port série
- Déclencheur de Base de Données
- Déclencheur TCP/IP
- Déclencheur HTTP
- Déclencheur Web Service

Extraction de données et Placement

Dès que l'impression est activée, le NiceLabel Automation extrait les données d'étiquette et les insère

dans les champs de variables du modèle d'étiquette.

Les Filtres d'extraction de données supportent :

- Fichiers de Texte Structurés
- Fichiers de Texte Non-Structurés
- Différents fichiers XML
- Données binaires : remplacement d'imprimante, export d'un logiciel d'origine, données d'appareils matériels, etc.

Exécution d'Action.

Quand les données ont été appliquées aux champs de variables correspondants sur les étiquettes, NiceLabel Automation effectue les actions. Les opérations de base comprennent généralement les actions **Ouvrir Étiquette** et **Imprimer Étiquette** pour imprimer les données extraites sur les étiquettes. Vous pouvez aussi envoyer les données à des endroits personnalisés, tels que des fichiers sur le disque, à un serveur Web, à des appareils matériels et beaucoup d'autres. En tout, vous avez le choix entre 30 actions différentes.

Voir plus d'informations concernant les Actions d'impression de base et avancée.

Conventions Typographiques

Le texte qui apparaît en **gras** se réfère aux noms et boutons du menu.

Le texte qui apparaît en *italique* se réfère aux options, actions de confirmation tel que Lecture seule et emplacements tels que Dossier.

Le texte encadré par les signes <Plus-petit et Plus-grand> Se réfère aux touches du clavier de l'ordinateur tel que `<Enter>`.

Les variables sont entourées de [crochets].

NOTE: Ceci représente la conception d'une note.

EXEMPLE: Ceci représente la conception d'un exemple.

Ceci représente la conception d'une meilleure pratique.

AVERTISSEMENT Ceci représente la conception d'un avertissement.

Ceci représente la conception d'un renseignement.

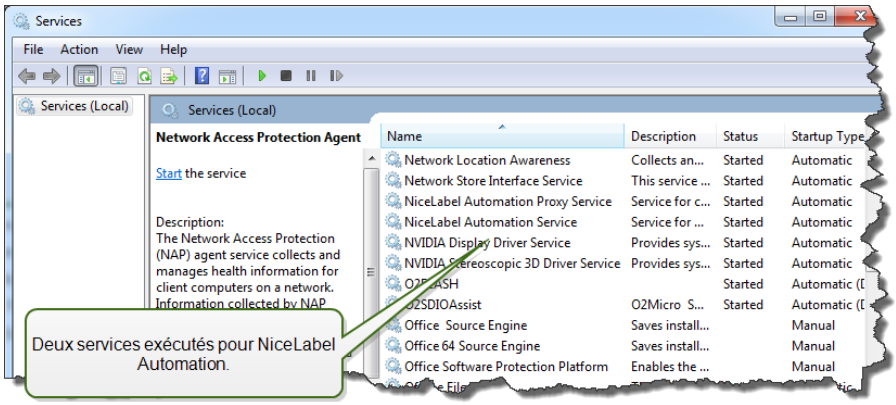
Paramétrage De L'Application

Architecture

NiceLabel Automation est une application basée sur un service. L'exécution de toutes les règles et actions est effectuée comme un processus d'arrière-plan avec les infos d'identification du compte d'utilisateur pour le Service.

Le NiceLabel Automation est composé de trois éléments.

- **Automation Builder.** Ceci est l'application de configuration que le concepteur utilisera pour créer les déclencheurs, filtres et actions à exécuter quand les données sont reçues par le déclencheur. Cette application s'exécute toujours comme une application 32-bits.
- **Automation Manager.** Ceci est l'application de gestion utilisée pour la surveillance de l'exécution des déclencheurs en temps réel et démarrer/arrêter les déclencheurs. Cette application s'exécute toujours comme une application 32-bits.
- **NiceLabel AutomationService.** Ceci est l'engin d'impression' qui exécule les règles définies dans les déclencheurs. Il y a en fait deux applications de service, le Service NiceLabel Automation et le Service Proxy NiceLabel Automation. Le Service détecte toujours le nombre de bits de la machine Windows et fonctionne au même niveau (par ex. en application 64-bits sous Windows 64-bits), alors que le Service Proxy sera toujours effectué comme un processus 32-bits.



Spécifications Système Requises

- CPU : Processeur Intel ou compatible de la famille x86
- Mémoire : 512 MB ou plus de RAM
- Disque dur : 1GB de place disponible

- Système d'exploitation : un des Systèmes d'exploitation Windows 32 ou 64 bits - Windows XP Service Pack 3, Windows Server 2003, Windows Server 2003 R2, Windows Vista, Windows Server 2008, Windows Server 2008 R2, Windows 7, Windows 8, Windows 8.1, Windows Server 2012, Windows Server 2012 R2
- Microsoft .NET Framework Version 4.0
- Écran : Résolution de l'écran 1024x768 ou plus élevée
- Concepteur d'étiquette :
  - Recommandé : NiceLabel Designer Pro ou NiceLabel PowerForms Desktop, chacun V6.0 ou supérieure
  - Minimum: NiceLabel Pro V5.4
- Pilotes d'imprimantes recommandés : NiceLabel Printer Drivers V5.1 ou supérieure
- Accès complet au 'Dossier Système' d'application, dans lequel les évènements sont sauvegardés dans une base de données.

  ```
  %ProgramData%\EuroPlus\NiceLabel Automation
  ```

- Accès complet aux dossier du compte utilisateurs service `%temp%`.

Installation

NOTE: Veuillez trouver ci-dessous une description sommaire de la procédure d'installation. Pour plus d'informations, consultez le **NiceLabel Automation Guide d'Installation**.

Avant de commencer l'installation, assurez-vous que votre infrastructure est compatible avec les Spécifications Système requises requises.

Pour installer NiceLabel Automation, effectuez les opérations suivantes :

1. Insérez le **NiceLabel Automation** DVD.
   Le menu principal de l'application démarre automatiquement.

   Si le menu principal de l'application ne démarre pas, effectuez un double-clic sur le fichier `START.EXE` du DVD.

2. Sélectionnez l'option **Installer NiceLabelProduits**.
   L'installation de NiceLabel Automation va démarrer.

3. Suivez les instructions de l'**Interface de paramétrage**.

   Durant l'installation, l'interface vous proposera l'introduction du nom d'utilisateur sous lequel le service NiceLabel Automation fonctionnera. Assurez-vous de sélectionner un nom d'utilisateur réel, car le service héritera les privilèges de ce nom d'utilisateur. Pour plus d'informations, consultez l'article Fonctionnement en Mode de Service.

Mise à niveau

Pour mettre NiceLabel Automation à niveau, installez la nouvelle version sur la version installée en la substituant. L'ancienne version sera enlevée durant la mise à niveau et remplacée par la nouvelle, en gardant les paramètres existants. La base de données du journal sera vidée durant la mise à niveau.

Activation

Vous devez activer le logiciel NiceLabel Automation pour permettre l'exécution des déclencheurs configurés. La procédure d'activation requiert une connexion Internet, de préférence sur la machine où vous installez le logiciel. La même procédure d'activation est utilisée pour activer la clé de licence d'essai.

NOTE: Vous pouvez activer le logiciel soit depuis Automation Builder ou Automation Manager et atteindre le même effet.

Activation dans Automation Builder

1. Exécuter **Automation Builder**.
2. Sélectionner **Fichiers>Outils>Gestion de Licence**.
   L'interface d'activation va démarrer.
3. Sélectionnez la méthode d'activation.
   - **Clé logicielle pour un utilisateur.** Dans ce cas, vous voulez activer NiceLabel Automation comme un serveur indépendant. Cliquez le bouton **Suivant** et suivez les instructions à l'écran.
   - **Serveur de licence Control Center.** Dans ce cas, vous voulez activer NiceLabel Automation depuis Control Center. Cliquez **Suivant** et sélectionnez le serveur Control Center qui a déjà la licence NiceLabel Automation activée. Référez-vous au Guide d'installation de Control Center pour les phases d'activation des produits dans Control Center.

Activation dans Automation Manager

1. Exécuter **Automation Manager**.
2. Allez à l'onglet **A propos de**.
3. Cliquez **Entrer la clé de licence**.
4. Sélectionnez la méthode d'activation.
   - **Clé logicielle pour un seul utilisateur.** Dans ce cas, vous voulez activer NiceLabel Automation comme un serveur indépendant. Cliquez le bouton **Suivant** et suivez les instructions à l'écran.
   - **Serveur de licence Control Center.** Dans ce cas, vous voulez activer NiceLabel Automation depuis Control Center. Cliquez **Suivant** et sélectionnez le serveur Control Center qui a déjà la licence NiceLabel Automation activée. Référez-vous au Guide d'installation de Control Center pour les phases d'activation des produits dans Control Center.

Activation sans accès Internet

Pour activer automatiquement NiceLabel Automation vous devez avoir une connexion Internet durant la procédure d'activation. Vous installez NiceLabel Automation sur le serveur sans connexion Internet mais vous aurez toutefois besoin d'un connexion Internet sur une autre machine, sur laquelle la procédure d'activation sera effectuée.

Suivez les instructions :

1. Suivez la procédure d'activation.
2. Introduisez la **Clé de licence**, le Numéro d'Inscription sera généré.
3. Cliquez le bouton **Sauvegarder les données d'inscription**.
4. Copiez le fichier sur une clé USB et allez à l'ordinateur qui a l'accès à Internet.
5. Ouvrez l'**URL** du fichier sauvegardé.
   La page d'activation Web va s'ouvrir.
6. Assurez-vous que les valeurs de tous les champs sont correctement introduites, ensuite cliquez le bouton **Activer**.
7. Mémorisez le Code d'Activation et introduisez-le sur le serveur avec NiceLabel Automation.
8. Cliquez le bouton **Terminer**.

Mode D'essai

Le mode d'essai vous permet de tester le produit NiceLabel Automation durant une période de 30 jours. Le mode d'essai vous donne les même fonctionnalités que la version avec licence, ce qui vous permet une évaluation du produit avant de l'acheter. Le Automation Manager affichera la notification d'essai en permanence ainsi que le nombre de jours restants pour l'essai. Quand le mode d'essai expire, le service NiceLabel Automation n'exécutera plus les déclencheurs. Le décompte des 30 jours commence le jour de l'installation.

NOTE: Vous pouvez étendre le mode d'essai en contactant votre revendeur NiceLabel et lui demandant une nouvelle clé d'essai. Vous devez activer la clé de licence d'essai. Pour plus d'informations, consultez l'article Activation.

Comprendre Les Filtres

Comprendre Les Filtres

NiceLabel Automation utilise des filtres pour définir la structure des données reçues par les déclencheurs. Chaque fois qu'un déclencheur reçoit une donnée, cette donnée est analysée par un ou plusieurs filtres qui extrait les valeurs nécessaires. Chaque filtre est configuré avec des règles décrivant la façon d'identifier les champs dans les données.

NOTE: Le résultat fourni par le filtre est une liste de champs et leurs valeurs (paires `champ-valeur`).

Types de Filtres

Pour plus d'informations, consultez les articles Configuration de Filtre de Texte Structuré, Configuration de Filtre de Données Non-structurées et Configuration de filtre XML.

Structure de Données

La complexité du filtre dépend de la structure des données. Les données qui sont déjà sous une forme structurée, telles que CSV ou XML peuvent être extraites facilement . Dans ce cas, les champs de données sont déjà définis dans les données. L'extraction des paires `nom-valeur` est rapide. Quand les données ne sont pas sous une structure claire, la définition des règles d'extraction prend plus de temps. Ces données peuvent se présenter sous forme de documents exportés et rapports d'un système original, de l'interception de communication entre deux appareils, une capture d'un flux d'impression et similaires.

NiceLabel Automation supporte différents types de données d'entrée qui peuvent toutes être analysées par un des types de filtres supportés. Vous devez choisir le filtre correct, qui correspond au type de données entrantes. Par exemple, vous utiliserez le **Filtre de Texte Structuré** pour des données entrantes CSV et vous utiliserez le **Filtre XML** pour les données entrantes XML. Pour toutes données non-structurées, vous utiliserez le **Filtre de Données Non-structurées**. Pour plus d'informations, consultez l'article Comprendre les Structures de Données.

Extraction de Données

Le filtre est seulement un ensemble de règle qui n'effectue aucune extraction par lui-même. Pour exécuter le filtre, lancez l'action Utilisation d'un Filtre de Données. L'action va exécuter les règles du filtre sur les valeurs et extraire les valeurs.

Chaque déclencheur peut exécuter autant d'actions "Utiliser un Filtre de Données" que nécessaires. Si vous recevez des données d'entrée composées qui ne peuvent pas être analysées par un seul filtre, vous pouvez définir et exécuter les règles de plusieurs filtres dans "Utiliser un Filtre de Données"

exécutés l'un après l'autre. A la fin, vous pouvez utiliser les valeurs extraites par toutes les actions pour la même étiquette.

Mapper le Champs aux Variables

Pour utiliser les valeurs extraites, vous devez les sauvegarder dans des variables. L'action "Utiliser un Filtre de Données" n'extrait pas seulement les valeurs, mais les sauve aussi dans des variables. Pour configurer ce processus, vous devez mapper la variable pour le champ respectif. La valeur du champ sera ensuite sauvegardée dans une variable mappée.

> Il est conseillé de définir les champs et variables avec les mêmes noms. Dans ce cas, l'élément d'auto-mappage va lier les variables aux champs de mêmes noms, éliminant le processus manuel.

L'auto-mappage est disponible pour tous les types de filtres supportés. Quand l'auto-mappage est activé, l'action "Utiliser un Filtre de Données" va extraire automatiquement les valeurs et les mapper aux variables de mêmes noms que les noms de champs. Pour plus d'informations, consultez les articles Activer la Structure Dynamique pour les filtres de Texte Structuré, Définition des Zones d'Affectation pour les filtres de Données Non-structurées et Définition de la Zone d'Affectation XML pour les filtres XML.

Définition des Actions à Exécuter pour les Données Extraites

Généralement vous allez exécuter des actions sur les données extraites, telles que **Ouvrir l'Étiquette**, **Imprimer l'Étiquette**, ou une des actions de connectivité de sortie. Il est extrêmement important d'incorporer vos actions dans l'action **Utiliser un Filtre de Données**. Ceci assurera que les actions incorporées sont exécutées pour chaque extraction de données.

> **EXEMPLE:** Si vous avez un fichier CSV de 5 lignes, l'action incorporée sera aussi exécutée 5 fois, une fois pour chaque extraction de données. Si les actions ne sont pas incorporées, elles ne seront effectuées qu'une seule fois et contiendront les données de la dernière extraction de données. Pour l'exemple ci-dessus, la cinquième ligne du CSV sera imprimée, mais pas les quatre premières lignes. Si vous utilisez des Sous-Zones, assurez-vous d'incorporer votre action sous l'espace réservé correct.

Configuration De Filtre De Texte Structuré

Filtre De Texte Structuré

Pour en savoir plus sur les filtres en général, consultez l'article Comprendre les Filtres.

Utilisez ce filtre chaque fois que vous recevez un fichier de texte structuré. Ce sont des fichiers de texte dans lesquels les champs sont identifiés par une des méthodes suivantes.

- **Les champs sont séparés par un caractère.** Généralement les séparateurs sont une virgule ou un double-point. Le fichier CSV (valeurs séparées par virgule) est un exemple typique.
- **Les champs contiennent un nombre fixe de caractères**. En d'autres termes, les champs sont définis par les colonnes de largeur fixe.

Si la première ligne contient les noms de champs, vous pouvez importer ces noms dans la configuration et sauter la phase de définition manuelle des noms de champs. Le filtre définira ensuite les règles des noms de champs et leurs longueurs (seulement dans le cas de colonnes de données à largeur fixe).

Définition de Structure

Vous avez les options suivantes pour définir la structure du fichier de texte.

- **Importation de la structure en utilisant l'Assistant de fichier texte.** Dans ce cas, cliquez le bouton **Importer la Structure de Données** dans le ruban et suivez les instructions à l'écran. Quand vous terminez l'Assistant, le type de base de données de texte et tous les champs seront définis. Si la première ligne de données contient les noms de champs, l'Assistant peut les importer. Ceci est la méthode recommandée si le déclencheur va toujours recevoir des données de structure identique.
- **Définition manuelle des champs.** Dans ce cas, vous devez définir le type de données manuellement (champs délimités ou de largeur de champ fixe et ensuite définir les noms de champs. Pour plus d'informations, consultez l'article Définition des Champs.
- **Lecture dynamique des champs.** Dans ce cas, le déclencheur peut recevoir des données de structures différentes, tel que des nouveaux noms de champs, et vous ne voulez pas mettre à jour le filtre pour chaque changement de structure. Le support dynamique va automatiquement lire tous les champs dans les données, même s'il existe des nouveaux champs ou certains anciens champs sont manquants. Pour plus d'informations, voir l'article Activer la Structure Dynamique.

La section Aperçu de Données simplifie la configuration. Le résultat des filtres définis est surligné dans la zone d'aperçu à chaque changement de configuration. Vous pouvez voir quelles données seront extraites pour chaque règle.

Définition Des Champs

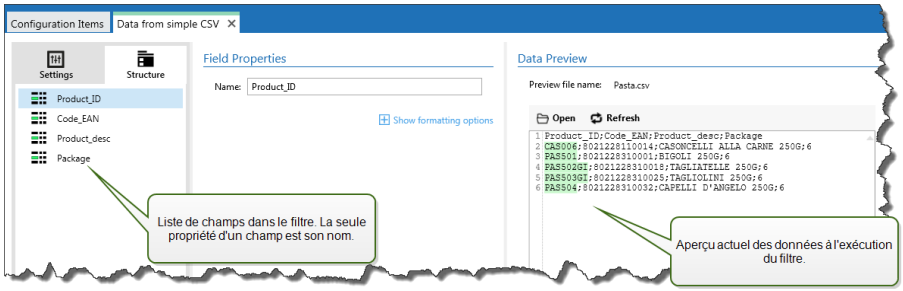
La définition des champs est très facile pour les fichier texte structurés. Vous avez deux options.

- **La délimitation définit les champs.** Dans ce cas, vous avez des séparateurs, tels que virgule ou double-point entre les champs. Vous devez seulement définir les noms de champs dans le même ordre qu'ils apparaitront dans les données reçues par le déclencheur.
- **Champs à largeur fixe.** Dans ce cas, vous devez seulement définir les noms de champs dans le même ordre qu'ils apparaitront dans les données reçues par le déclencheur et définir le nombre de caractères que le champ va occuper. Autant de caractères seront lus depuis les données pour ce champ.

Aperçu des données

Cette section fournit un aperçu de la définition de champ. Quand l'élément défini est sélectionné, l'aperçu va surligner son emplacement dans les données aperçues.

- **Aperçu nom de fichier.** Spécifie le fichier qui contient l'échantillon de données qui sera analysé dans le filtre. Le fichier d'aperçu est copié de la définition du filtre. Si vous changez le nom du fichier d'aperçu, le nouveau nom de fichier sera sauvegardé.
- **Ouvrir.** Sélectionne un autre fichier selon lequel vous voulez exécuter les règles du filtre.
- **Rafraîchir.** Relance les règles du filtre en fonction du contenu du nom du fichier d'aperçu. La section Aperçu de Données sera mise à jour avec le résultat.



Options de Formatage

Cette section définit les fonctions de manipulation de chaîne de caractères qui seront appliquées aux variables ou champs sélectionnés. Vous pouvez sélectionner une ou plusieurs fonctions. Les fonctions seront appliquées dans l'ordre sélectionné dans l'interface utilisateur, de haut en bas.

- **Enlever les espaces au début.** Enlève tous les caractères d'espacement (code décimal ASCII 32) du début de la chaîne de caractères.
- **Enlever les espaces à la fin.** Enlève tous les caractères d'espacement (code décimal ASCII 32) à la fin de la chaîne de caractères.
- **Enlever les caractères d'ouverture et fermeture.** Efface la première occurrence du caractère d'ouverture et de fermeture trouvé dans la chaîne de caractères.

  EXEMPLE: Si vous utilisez "{" comme caractère d'ouverture et "}" comme caractère de fermeture, la chaîne d'entrée `{{selection}}` sera convertie en `{selection}`.

- **Rechercher et remplacer.** Exécute une recherche classique et remplace la fonction selon la valeur fournie pour *find what* et *replace with* (chercher-quoi et remplacer-par). Vous pouvez utiliser des expressions classiques.

  NOTE: Il y a plusieurs implémentations des expressions classiques utilisées. NiceLabel Automation utilise la syntaxe .NET Framework pour les expressions classiques. Pour plus d'informations, consultez la Base de Connaissances article KB250.

- **Remplacer les caractères non-imprimables par un espace.** Remplace tous les caractères de contrôle de la chaîne de caractères par des caractères d'espacement (code décimal ASCII 32). Les caractères non-imprimables sont des caractères à valeurs ASCII entre 0-31 et 127-159.
- **Effacer les caractères non-imprimables.** Efface tous les caractères dans la chaîne de caractères. Les caractères non-imprimables sont des caractères à valeurs ASCII entre 0-31 et 127-159.
- **Décoder les caractères spéciaux.** Les caractères spéciaux (ou caractères de contrôle) sont des caractères qui ne sont pas disponibles sur le clavier, tels que Retour Charriot et Passage à la Ligne. NiceLabel Automation utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Charriot et <LF> pour Passage à la Ligne. Pour plus d'informations, consultez l'article Introduction de Caractères Spéciaux (Codes de Contrôle). Cette option convertit les caractères spéciaux de la syntaxe NiceLabel en caractères binaires réels.

  EXEMPLE: Quand vous recevez les données "<CR><LF>", NiceLabel Automation les utilisera comme une chaîne complète de 8 caractères. Pour interpréter et utiliser

les données comme deux caractères binaires `CR` (Retour Charriot - code ASCII 13) et `LF` (Passage à la Ligne - code ASCII 10), vous devrez activer cette nouvelle option.

- **Rechercher et effacer tout avant.** Trouve la chaîne de caractères fournie et efface tous les caractères du début des données jusqu'à la chaîne de caractères. La chaîne de caractères trouvée peut aussi être effacée.
- **Rechercher et effacer tout après.** Trouve la chaîne de caractères fournie et efface tous les caractères depuis la chaîne de caractères jusqu'à la fin des données. La chaîne de caractères trouvée peut aussi être effacée.

Activer La Structure Dynamique

Le filtre de Texte Structuré à la possibilité d'identifier automatiquement les champs et leurs valeurs, ce qui élimine la nécessité de mappage *variable à champ*.

Cette fonctionnalité est utile quand le déclencheur reçoit la donnée de la structure changeable. La structure principale de données est identique, par ex. les champs sont délimités par une virgule, ou la même structure XML, mais **l'ordre** dans lequel les champs sont représentés est changé et/ou **le nombre de champs** a changé; il peut y avoir de nouveaux champs, ou certains champs ne sont plus disponibles. La structure sera automatiquement identifiée par le filtre. En même temps, les noms et valeurs de champs (paires `nom:valeur`) seront lues des données, éliminant la nécessité de mapper les variables manuellement.

L'action Utilisation d'un Filtre de Données n'affichera pas de possibilité de mappage, car le mappage se fera de façon dynamique. Vous ne devez même as définir les variables d'étiquettes dans la configuration du déclencheur. L'action assignera les valeurs de champs aux variables d'étiquettes de même nom sans avoir besoin des variables importées de l'étiquette. Toutefois, la règle s'applique seulement à l'action Impression d'étiquette. Si vous voulez utiliser les valeurs de champs dans une autre action, vous devrez définir les variables dans le déclencheur, tout en conservant le mappage automatique *variable-à-champ*.

NOTE: Aucune erreur ne sera signalée si le champ disponible dans les données d'entrée n'a pas de variable d'étiquette correspondant. Les variables manquantes sont ignorées en silence.

Configuration de la structure dynamique

Pour configurer la structure dynamique, activez l'option **Structure Dynamique** dans les caractéristiques des Filtres de Textes Structurés.

- La première ligne de données doit contenir les noms de champs.
- La ligne que vous sélectionnez pour **Commencer l'importation à la ligne** doit être la ligne contenant les noms de champs (généralement la première ligne dans les données).
- La structure de données doit être délimitée.
- Vous pouvez formater les données, si nécessaire.

  Options de Formatage

  Cette section définit les fonctions de manipulation de chaîne de caractères qui seront appliquées aux variables ou champs sélectionnés. Vous pouvez sélectionner une ou

plusieurs fonctions. Les fonctions seront appliquées dans l'ordre sélectionné dans l'interface utilisateur, de haut en bas.

- **Enlever les espaces au début.** Enlève tous les caractères d'espacement (code décimal ASCII 32) du début de la chaîne de caractères.
- **Enlever les espaces à la fin.** Enlève tous les caractères d'espacement (code décimal ASCII 32) à la fin de la chaîne de caractères.
- **Enlever les caractères d'ouverture et fermeture.** Efface la première occurrence du caractère d'ouverture et de fermeture trouvé dans la chaîne de caractères.

  EXEMPLE: Si vous utilisez "{" comme caractère d'ouverture et "}" comme caractère de fermeture, la chaîne d'entrée `{{selection}}` sera convertie en `{selection}`.

- **Rechercher et remplacer.** Exécute une recherche classique et remplace la fonction selon la valeur fournie pour *find what* et *replace with* (chercher-quoi et remplacer-par). Vous pouvez utiliser des expressions classiques.

  NOTE: Il y a plusieurs implémentations des expressions classiques utilisées. NiceLabel Automation utilise la syntaxe .NET Framework pour les expressions classiques. Pour plus d'informations, consultez la Base de Connaissances article KB250.

- **Remplacer les caractères non-imprimables par un espace.** Remplace tous les caractères de contrôle de la chaîne de caractères par des caractères d'espacement (code décimal ASCII 32). Les caractères non-imprimables sont des caractères à valeurs ASCII entre 0-31 et 127-159.
- **Effacer les caractères non-imprimables.** Efface tous les caractères dans la chaîne de caractères. Les caractères non-imprimables sont des caractères à valeurs ASCII entre 0-31 et 127-159.
- **Décoder les caractères spéciaux.** Les caractères spéciaux (ou caractères de contrôle) sont des caractères qui ne sont pas disponibles sur le clavier, tels que Retour Charriot et Passage à la Ligne. NiceLabel Automation utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Charriot et <LF> pour Passage à la Ligne. Pour plus d'informations, consultez l'article Introduction de Caractères Spéciaux (Codes de Contrôle).
  Cette option convertit les caractères spéciaux de la syntaxe NiceLabel en caractères binaires réels.

  EXEMPLE: Quand vous recevez les données "<CR><LF>", NiceLabel Automation les utilisera comme une chaîne complète de 8 caractères. Pour interpréter et utiliser les données comme deux caractères binaires `CR` (Retour Charriot - code ASCII 13) et `LF` (Passage à la Ligne - code ASCII 10), vous devrez activer cette nouvelle option.

- **Rechercher et effacer tout avant.** Trouve la chaîne de caractères fournie et efface tous les caractères du début des données jusqu'à la chaîne de caractères. La chaîne de caractères trouvée peut aussi être effacée.

- **Rechercher et effacer tout après.** Trouve la chaîne de caractères fournie et efface tous les caractères depuis la chaîne de caractères jusqu'à la fin des données. La chaîne de caractères trouvée peut aussi être effacée.

Configuration De Filtre De Données Non-structurées

Filtre De Données Non-structurées

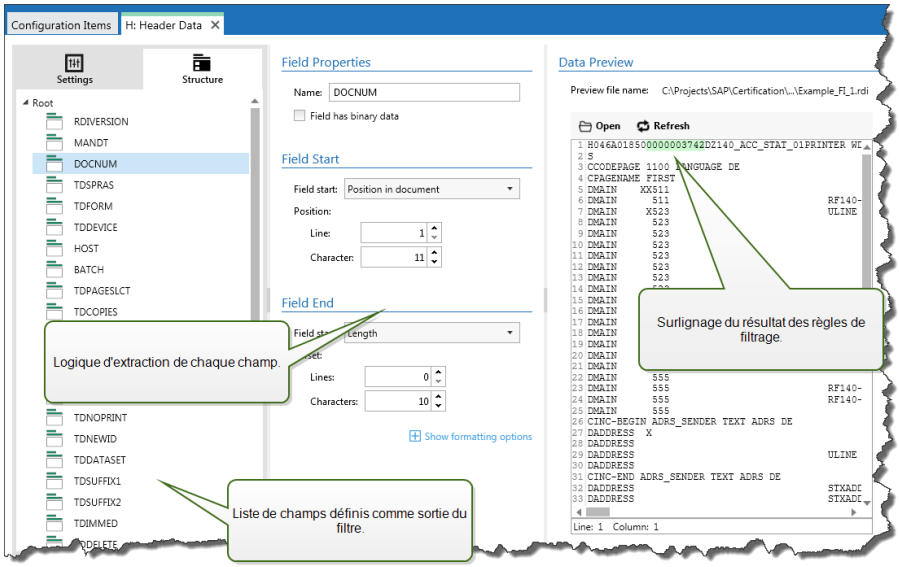
Pour en savoir plus sur les filtres en général, consultez l'article Comprendre les Filtres.

Utilisez ce filtre chaque fois que le déclencheur reçoit des données non-structurées, telles que des documents et rapports exportés de systèmes originaux, l'interception d'une communication entre des appareils, une capture de flux d'impression et similaires. Le filtre vous permet d'extraire les champs individuels, les champs répétables dans les sous-zones, et même les paires `nom-valeur`.

Les éléments que vous pouvez utiliser pour configurer le filtre :

- **Champ.** Spécifie l'emplacement des données de champ entre l'emplacement du début et de fin du champ. Il y a différentes options pour définir l'emplacement du champ, depuis le codage-en-dur de la position pour permettre les placements relatifs. Vous devez mapper les champs définis aux variables respectives dans l'action Utilisation d'un Filtre de Données. Pour plus d'informations, consultez l'article Définition des Champs.
- **Sous-zone.** Spécifie l'emplacement des données répétables. Chaque sous-zone définit au moins un bloc de données, qui à son tour contiendra des données pour les étiquettes. Il peut y avoir des sous-zones définies dans les sous-zones, permettant la définition de structures complexes. Vous pouvez définir des champs dans chaque bloc de données. Vous devez mapper les champs définis aux variables respectives dans l'action Utilisation d'un Filtre de Données. Pour chaque sous-zone, un nouveau niveau d'espace réservé sera défini dans "Utiliser le Filtre de Données", vous permettant de mapper les variables aux champs de ce niveau. Pour plus d'informations, consultez l'article Définition de sous-zones.
- **Zone d'affectation.** Spécifie l'emplacement des données répétables contenant les paires `nom-valeur`. Les noms de champs et leurs valeurs sont lus simultanément. Le mappage aux variables est fait automatiquement. Utilisez cet élément pour adapter le filtre aux données d'entrée changeables, éliminant le temps de maintenance. La zone d'affectation peut être définie au niveau racine du document, ou dans la sous-zone. Pour plus d'informations, consultez l'article Définition des Zones d'Affectation.

La section Aperçu de Données simplifie la configuration. Le résultat des filtres définis est surligné dans la zone d'aperçu à chaque changement de configuration. Vous pouvez voir quelles données seront extraites pour chaque règle.



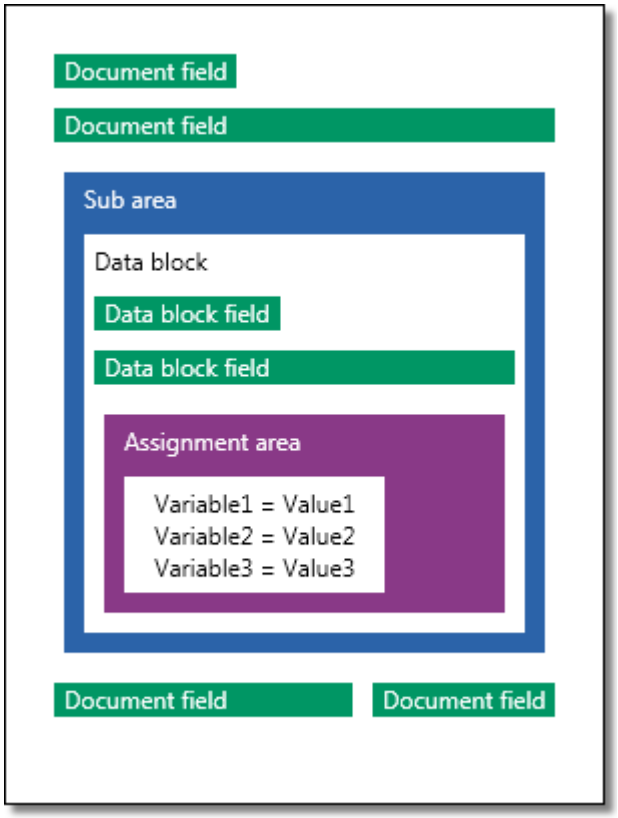
Général

Cette section définit les caractéristiques générales du filtre de données non-structurées.

- **Nom.** Spécifie le nom du filtre. Utilisez un nom descriptif qui identifiera ce que fait le filtre. Vous pouvez le changer à tout instant.
- **Description.** Procure la possibilité de décrire la fonctionnalité de ce filtre. Vous pouvez l'utiliser pour écrire une courte description de la fonction du filtre.
- **Encoding. (encodage)** Spécifie l'encodage des données avec lesquelles ce filtre va travailler.
- **Ignorer les lignes vides dans les blocs de données.** Spécifie de ne pas signaler d'erreur si le filtre extrait des valeurs de champs vides des blocs de données.

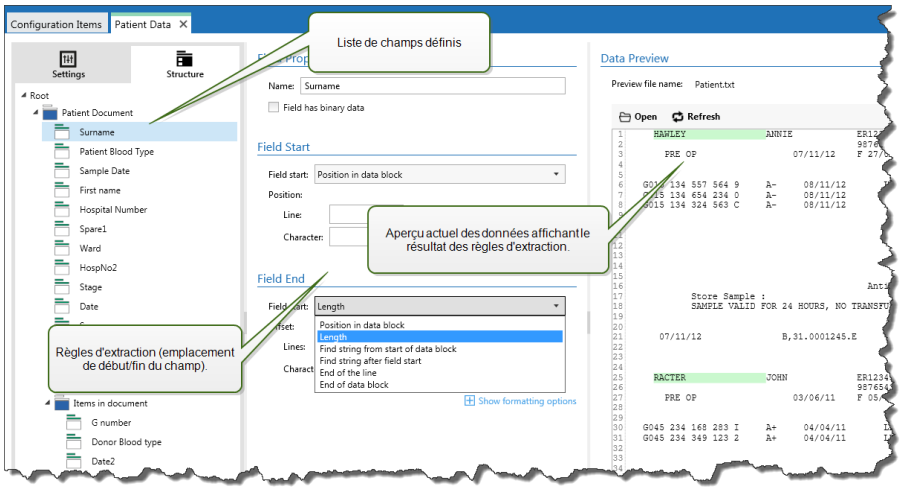
Exemple

Ce champ peut être défini au niveau racine comme champ de document. Les champs peuvent être définis dans les blocs de données. Les paires `nom-valeur` peuvent être définies dans la zone d'affectation.



Définition Des Champs

Quand vous définissez un champ, vous devez définir son nom et un règle d'extraction de la valeur du champ des données. Quand le filtre sera exécuté, les règles d'extraction seront appliquées aux données d'entrée et les résultat assignés aux champs.



Caractéristiques de Champ

- **Nom.** Spécifie le nom de champ unique.
- **Champ contenant des données binaires.** Spécifie que le champ contiendra des données binaires. Ne l'activez que si vous attendez vraiment de recevoir des données binaires.

Début de champ

- **Position dans le document.** Me point de début/fin est déterminé par la position codée-en-dur

dans les données. L'origine des coordonnées est le coin en haut à gauche. Le caractère dans la position déterminée est inclut dans le données extraites.

- **Fin du document.** Le point de début/fin est à la fin du document. Vous pouvez aussi définir un décalage depuis la fin pour un nombre de lignes et/ou de caractères.
- **Trouver une chaîne de caractères depuis le début du document.** Le point de début/fin est défini par la position de recherche de chaîne de caractères. Quand la chaîne de caractères désirée est trouvée, le caractère suivant détermine le point de début/fin. La chaîne de caractères recherchée n'est pas inclue dans les données extraites. La recherche par défaut est sensible à la casse.
    - **Commencer la recherche depuis une position absolue.** Vous pouvez affiner la recherche en changeant la position de départ de début-donnée (position 1,1) à un décalage. Utilisez cette fonctionnalité pour sauter la recherche au début des données.
    - **Occurrence.** Spécifie l'occurrence de la chaîne de caractères à retrouver. Utilisez cette option si vous n'attendez pas pour déterminer la position début/fin après la première chaîne de caractères trouvée.
    - **Décalage depuis la chaîne de caractères.** Spécifie le décalage positif ou négatif après la chaîne de caractères recherchée.

      EXEMPLE: Le décalage défini sera inclut dans la recherche de chaîne de caractères avec les données extraites.

Fin de champ

- **Position dans le document.** Me point de début/fin est déterminé par la position codée-en-dur dans les données. L'origine des coordonnées est le coin en haut à gauche. Le caractère dans la position déterminée est inclut dans le données extraites.
- **Fin du document.** Le point de début/fin est à la fin du document. Vous pouvez aussi définir un décalage depuis la fin pour un nombre de lignes et/ou de caractères.
- **Trouver une chaîne de caractères depuis le début du document.** Le point de début/fin est défini par la position de recherche de chaîne de caractères. Quand la chaîne de caractères désirée est trouvée, le caractère suivant détermine le point de début/fin. La chaîne de caractères recherchée n'est pas inclue dans les données extraites. La recherche par défaut est sensible à la casse.
    - **Commencer la recherche depuis une position absolue.** Vous pouvez affiner la recherche en changeant la position de départ de début-donnée (position 1,1) à un décalage. Utilisez cette fonctionnalité pour sauter la recherche au début des données.
    - **Occurrence.** Spécifie l'occurrence de la chaîne de caractères à retrouver. Utilisez cette option si vous n'attendez pas pour déterminer la position début/fin après la première chaîne de caractères trouvée.
    - **Décalage depuis la chaîne de caractères.** Spécifie le décalage positif ou négatif après la chaîne de caractères recherchée.

      EXEMPLE: Le décalage défini sera inclut dans la recherche de chaîne de caractères avec les données extraites.

- **Trouver une chaîne de caractères après le début du champ.** Le point de début/fin est défini par la position de la recherche-de-chaîne de caractères comme dans l'option **Recherche de chaîne depuis le début du document**, mais la recherche commence après la position du champ/zone, pas au début des données.
- **Longueur.** Spécifie la longueur des données dans les lignes en caractères. Le nombre de lignes et/ou caractères spécifié sera extrait depuis la position de départ.
- **Fin de ligne.** Spécifie d'extraire les données depuis la position de départ à la fin de la même ligne. Vous pouvez définir un décalage négatif depuis la fin de ligne.

Options de Formatage

Cette section définit les fonctions de manipulation de chaîne de caractères qui seront appliquées aux variables ou champs sélectionnés. Vous pouvez sélectionner une ou plusieurs fonctions. Les fonctions seront appliquées dans l'ordre sélectionné dans l'interface utilisateur, de haut en bas.

- **Enlever les espaces au début.** Enlève tous les caractères d'espacement (code décimal ASCII 32) du début de la chaîne de caractères.
- **Enlever les espaces à la fin.** Enlève tous les caractères d'espacement (code décimal ASCII 32) à la fin de la chaîne de caractères.
- **Enlever les caractères d'ouverture et fermeture.** Efface la première occurrence du caractère d'ouverture et de fermeture trouvé dans la chaîne de caractères.

  EXEMPLE: Si vous utilisez "{" comme caractère d'ouverture et "}" comme caractère de fermeture, la chaîne d'entrée `{{selection}}` sera convertie en `{selection}`.

- **Rechercher et remplacer.** Exécute une recherche classique et remplace la fonction selon la valeur fournie pour *find what* et *replace with* (chercher-quoi et remplacer-par). Vous pouvez utiliser des expressions classiques.

  NOTE: Il y a plusieurs implémentations des expressions classiques utilisées. NiceLabel Automation utilise la syntaxe .NET Framework pour les expressions classiques. Pour plus d'informations, consultez la Base de Connaissances article KB250.

- **Remplacer les caractères non-imprimables par un espace.** Remplace tous les caractères de contrôle de la chaîne de caractères par des caractères d'espacement (code décimal ASCII 32). Les caractères non-imprimables sont des caractères à valeurs ASCII entre 0-31 et 127-159.
- **Effacer les caractères non-imprimables.** Efface tous les caractères dans la chaîne de caractères. Les caractères non-imprimables sont des caractères à valeurs ASCII entre 0-31 et 127-159.
- **Décoder les caractères spéciaux.** Les caractères spéciaux (ou caractères de contrôle) sont des caractères qui ne sont pas disponibles sur le clavier, tels que Retour Charriot et Passage à la Ligne. NiceLabel Automation utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Charriot et <LF> pour Passage à la Ligne. Pour plus d'informations, consultez l'article Introduction de Caractères Spéciaux (Codes de Contrôle). Cette option convertit les caractères spéciaux de la syntaxe NiceLabel en caractères binaires réels.

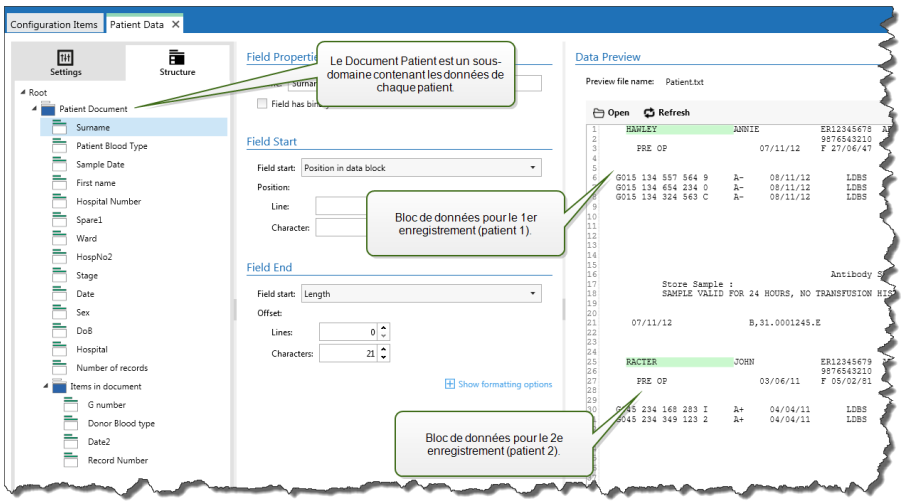
EXEMPLE: Quand vous recevez les données "<CR><LF>", NiceLabel Automation les utilisera comme une chaîne complète de 8 caractères. Pour interpréter et utiliser les données comme deux caractères binaires `CR` (Retour Charriot - code ASCII 13) et `LF` (Passage à la Ligne - code ASCII 10), vous devrez activer cette nouvelle option.

- **Rechercher et effacer tout avant.** Trouve la chaîne de caractères fournie et efface tous les caractères du début des données jusqu'à la chaîne de caractères. La chaîne de caractères trouvée peut aussi être effacée.
- **Rechercher et effacer tout après.** Trouve la chaîne de caractères fournie et efface tous les caractères depuis la chaîne de caractères jusqu'à la fin des données. La chaîne de caractères trouvée peut aussi être effacée.

Définition De Sous-zones

Une sous-zone est la section de données dans laquelle plusieurs blocs de données sont identifiés par la même règle d'extraction. Chaque bloc de données fournit les données pour une seule étiquette. Tous les blocs de données doivent être identifiés par la même règle de configuration. Chaque bloc de données peut contenir une autre sous-zone. Vous pouvez définir un nombre illimité de sous-zones imbriquées dans les sous-zones apparentées.

Quand le filtre contient la définition d'une sous-zone, l'action Utilisation d'un Filtre de Données affichera les sous-zones avec les espaces réservés imbriqués. Toute action imbriquée sous une telle sous-zone s'exécutera seulement pour les blocs de données à ce niveau. Vous pouvez imprimer différentes étiquettes avec des données de différentes sous-zones.



Configuration des Sous-zones

La sous-zone est configurée par des règles similaires que pour les champs individuels. Chaque sous-zone est définie par les paramètres suivants.

- **Nom de la sous-zone.** Spécifie le nom de la sous-zone.
- **Blocs de données.** Spécifie la façon d'identifier les blocs de données dans la sous-zone Chaque sous-zone contient au moins un bloc de données. Chaque bloc de données fournit les données pour une seule étiquette.

- **Chaque bloc contient un nombre de lignes fixe.** Spécifie que chaque bloc de données dans une sous-zone contient le nombre de lignes fixe fourni. Utilisez cette option si vous savez que chaque bloc de données contient exactement le même nombre de lignes.
- **Les Blocs commences par une chaîne de caractères.** Spécifie que les blocs de données commencent par la chaîne de caractères fournie. Tout le contenu entre deux chaînes de caractères appartient à un bloc de données séparé. Le contenu entre la dernière chaîne de caractères et la fin des données identifie le dernier bloc de données.
- **Le Bloc se termine par une chaîne de caractères.** Spécifie que les blocs de données se terminent par la chaîne de caractères fournie. Tout le contenu entre deux chaînes de caractères appartient à un bloc de données séparé. Le contenu entre le début des données et la première chaîne de caractères identifie le premier bloc de données.
- **Les Blocs sont séparés par une chaîne de caractères.** Spécifie que les blocs de données sont séparés par la chaîne de caractères fournie. Tout le contenu entre deux chaînes de caractères fournies appartient à un bloc de données séparé.

- **Début du Premier Bloc de Données.** Spécifie la position de départ du premier bloc de données et donc la position de départ de la sous-zone. Généralement, la position de départ est le début des données reçues. Les paramètres de configuration sont les mêmes que pour la définition des champs. Pour plus d'informations, consultez l'article Définition des Champs.
- **Fin du Dernier Bloc de Données.** Spécifie la position de fin du dernier bloc de données et donc la position de fin de la sous-zone. Généralement, la position de fin est la fin des données reçues. Les paramètres de configuration sont les mêmes que pour la définition des champs. Pour plus d'informations, consultez l'article Définition des Champs.

Configuration des Champs Dans la Sous-zone

Les champs dans la sous-zone sont configurés en utilisant les mêmes paramètres que pour les champs définis au niveau racine. Pour plus d'informations, consultez l'article Définition des Champs.

NOTE: Les numéros des lignes de champs se réfèrent à la position dans le bloc de données, pas la position dans les données d'entrée.

Aperçu des données

Cette section fournit un aperçu de la définition de champ. Quand l'élément défini est sélectionné, l'aperçu va surligner son emplacement dans les données aperçues.

- **Aperçu nom de fichier.** Spécifie le fichier qui contient l'échantillon de données qui sera analysé dans le filtre. Le fichier d'aperçu est copié de la définition du filtre. Si vous changez le nom du fichier d'aperçu, le nouveau nom de fichier sera sauvegardé.
- **Ouvrir.** Sélectionne un autre fichier selon lequel vous voulez exécuter les règles du filtre.
- **Rafraîchir.** Relance les règles du filtre en fonction du contenu du nom du fichier d'aperçu. La section Aperçu de Données sera mise à jour avec le résultat.

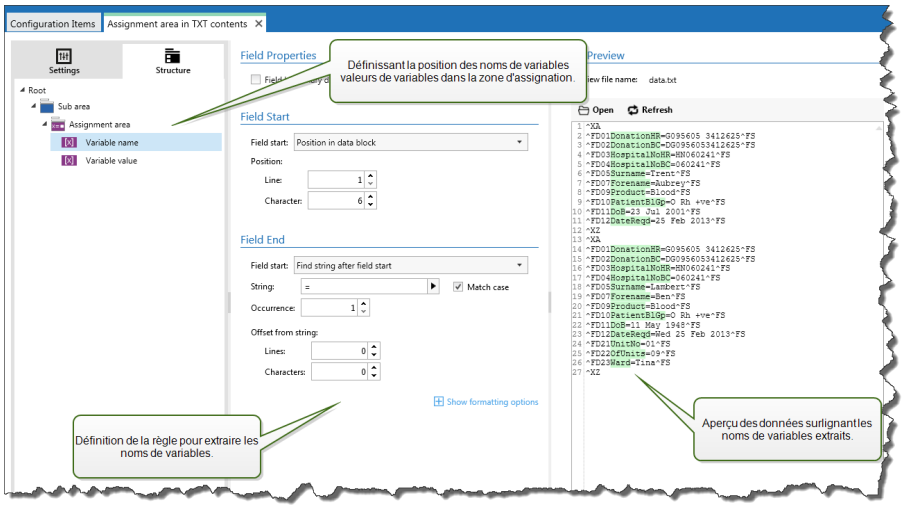
Définition Des Zones D'Affectation

Le filtre de Texte Non-Structuré a la possibilité d'identifier automatiquement les champs et leurs valeurs dans les données, ce qui élimine la nécessité de mappage *variable à champ*.

Cette fonctionnalité est utile quand le déclencheur reçoit la donnée de la structure changeable. La structure principale de données est identique, par ex. les champs sont délimités par une virgule, ou la même structure XML, mais **l'ordre** dans lequel les champs sont représentés est changé et/ou **le nombre de champs** a changé; il peut y avoir de nouveaux champs, ou certains champs ne sont plus disponibles. La structure sera automatiquement identifiée par le filtre. En même temps, les noms et valeurs de champs (paires `nom:valeur`) seront lues des données, éliminant la nécessité de mapper les variables manuellement.

L'action Utilisation d'un Filtre de Données n'affichera pas de possibilité de mappage, car le mappage se fera de façon dynamique. Vous ne devez même as définir les variables d'étiquettes dans la configuration du déclencheur. L'action assignera les valeurs de champs aux variables d'étiquettes de même nom sans avoir besoin des variables importées de l'étiquette. Toutefois, la règle s'applique seulement à l'action Impression d'étiquette. Si vous voulez utiliser les valeurs de champs dans une autre action, vous devrez définir les variables dans le déclencheur, tout en conservant le mappage automatique *variable-à-champ*.

NOTE: Aucune erreur ne sera signalée si le champ disponible dans les données d'entrée n'a pas de variable d'étiquette correspondant. Les variables manquantes sont ignorées en silence.



Configuration de la Zone d'Affectation

La zone d'affectation est configurée en utilisant la même procédure que la sous-zone. Pour plus d'informations, consultez l'article Définition de sous-zones. La zone d'affectation peut être définie au niveau des données racine, apparaissant une seule fois. Ou elle peut être configurée dans une sous-zone, elle sera ainsi exécutée pour chaque bloc de données dans la sous-zone.

Configuration des Champs dans la Zone d'Affectation

Quand vous créez la zone d'affectation, le filtre définira automatiquement deux espaces réservés, qui définiront les paires `nom-valeur`.

- **Nom de variable.** Spécifie le champ dont le contenu sera le nom de variable (composant `nom` en paire). Configurez le champ en utilisant la même procédure que pour les champs de document. Pour plus d'informations, consultez l'article Définition des Champs.

- **Valeur de variable.** Spécifie le champ dont le contenu sera le nom de variable (composant `valeur` en paire). Configurez le champ en utilisant la même procédure que pour les champs de document. Pour plus d'informations, consultez l'article Définition des Champs.

Exemple

La zone entre ^XA et ^XZ est la zone d'affectation. Chaque ligne dans la zone l'affectation fournit les paires `nom-valeur`. Le nom est défini comme une valeur entre le 6e caractère de la ligne et le signe égal. La valeur est définie comme la valeur entre le signe égal et la fin de la ligne, avec un décalage négatif de trois caractères

```
^XA
^FD01DonationHR=G095605 3412625^FS
^FD02DonationBC=DG0956053412625^FS
^FD03HospitalNoHR=HN060241^FS
^FD04HospitalNoBC=060241^FS
^FD05Surname=Hawley^FS
^FD07Forename=Annie^FS
^FD09Product=Blood^FS
^FD10PatientBlGp=O Rh +ve^FS
^FD11DoB=27 June 1947^FS
^FD12DateReqd=25 Dec 2012^FS
^XZ
```

Plus plus d'informations, consultez l'article Exemples.

Configuration De Filtre XML

Filtre XML

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Pour en savoir plus sur les filtres en général, consultez l'article Comprendre les Filtres.

Utilisez ce filtre chaque fois que le déclencheur reçoit des données encodées XML. Le filtre vous permet d'extraire les champs individuels, les champs répétables dans les sous-zones, et même les paires `nom-valeur`. La structure XML définit les éléments et sous-éléments, attributs et leurs valeurs et les valeurs texte (valeurs d'élément).

Vous pouvez définir la structure du fichier XML vous-même, mais il est conseillé d'importer la structure du modèle de fichier XML existant. Cliquez le bouton **Importer la Structure de Données** dans le ruban. Quand vous importez la structure XML, la section Aperçu de Données affichera le contenu XML et va surligner les éléments et attributs que vous définissez comme champs de sortie.

Pour utiliser les éléments XML, vous devez les configurer comme suit :

- **Valeur de variable.** Spécifie que vous voulez utiliser les éléments sélectionnés comme champs et allez mapper leurs valeurs aux variables respectives dans l'action Utilisation d'un Filtre de Données. Pour plus d'informations, consultez l'article Définition des Champs XML.
- **Blocs de données.** Spécifie que l'élément sélectionné survient plusieurs fois et fournira des données pour une seule étiquette. Le bloc de données peut être défini comme zone répétable, comme zone d'affectation ou les deux.

- **Zone répétable.** Spécifie que vous voulez extraire les valeurs de tous les blocs de données répétables, pas seulement le premier. Vous pouvez définir des champs dans chaque bloc de données. Vous devez mapper les champs définis aux variables respectives dans l'action Utilisation d'un Filtre de Données. Pour plus d'informations, consultez l'article Définition des Éléments Répétables.
- **Zone d'affectation.** Spécifie que le bloc de données contient des paires `nom-valeur`. Les noms de champs et leurs valeurs sont lus simultanément. Le mappage aux variables est fait automatiquement. Utilisez cet élément pour adapter le filtre aux données d'entrée changeables, éliminant le temps de maintenance. Pour plus d'informations, consultez l'article Définition de la Zone d'Affectation XML.

La section Aperçu de Données simplifie la configuration. Le résultat des filtres définis est surligné dans la zone d'aperçu à chaque changement de configuration. Vous pouvez voir quelles données seront extraites pour chaque règle.

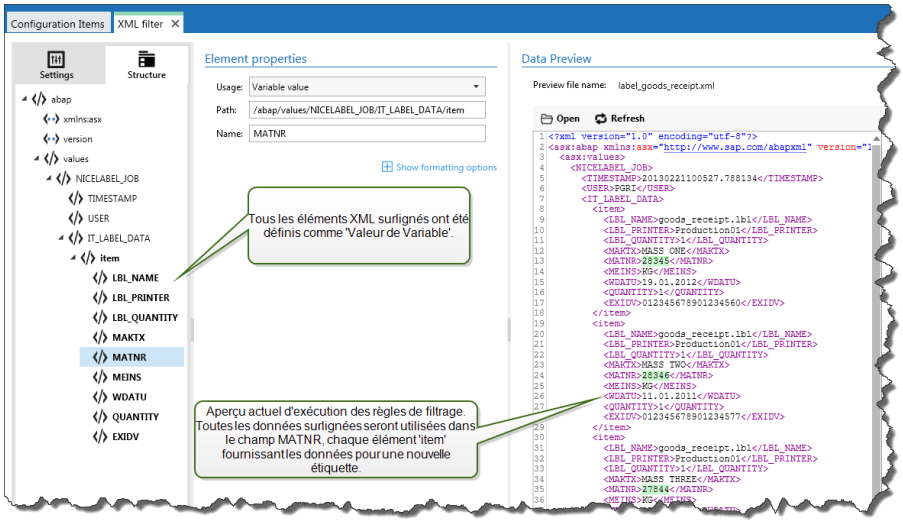
Définition Des Champs XML

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Quand vous définissez le champ XML, vous rendez disponible la valeur de l'élément sélectionné comme un champ. La définition du filtre va fournir un tel champ pour le mappage à la variable dans l'action Utilisation d'un Filtre de Données. Vous pouvez extraire la valeur de l'élément ou la valeur de l'attribut.

Pour définir la valeur de l'élément comme un champ, effectuez les opérations suivantes :

1. Sélectionnez l'élément ou attribut dans la liste de structure.
2. Pour **Usage** sélectionnez **Valeur de Variable**.
3. L'élément dans la liste de structure sera affiché en caractères gras, indiquant qu'il est utilisé.
4. L'élément ou nom d'attribut sera utilisé comme le nom du champ de sortie.
5. La section Aperçu de Données surlignera la valeur de l'élément sélectionné.


Configuration Items
XML filter
Settings
Structure
Element properties
Usage: Variable value
Path: /abap/values/NICELABEL_JOB/IT_LABEL_DATA/item
Name: MATNR
Show formatting options
Data Preview
Preview file name: label_goods_receipt.xml
Open
Refresh
Tous les éléments XML surlignés ont été définis comme 'Valeur de Variable'.
Aperçu actuel d'exécution des règles de filtrage. Toutes les données surlignées seront utilisées dans le champ MATNR, chaque élément 'item' fournissant les données pour une nouvelle étiquette.

Options de Formatage

Cette section définit les fonctions de manipulation de chaîne de caractères qui seront appliquées aux

variables ou champs sélectionnés. Vous pouvez sélectionner une ou plusieurs fonctions. Les fonctions seront appliquées dans l'ordre sélectionné dans l'interface utilisateur, de haut en bas.

- **Enlever les espaces au début.** Enlève tous les caractères d'espacement (code décimal ASCII 32) du début de la chaîne de caractères.
- **Enlever les espaces à la fin.** Enlève tous les caractères d'espacement (code décimal ASCII 32) à la fin de la chaîne de caractères.
- **Enlever les caractères d'ouverture et fermeture.** Efface la première occurrence du caractère d'ouverture et de fermeture trouvé dans la chaîne de caractères.

  EXEMPLE: Si vous utilisez "{" comme caractère d'ouverture et "}" comme caractère de fermeture, la chaîne d'entrée `{{selection}}` sera convertie en `{selection}`.

- **Rechercher et remplacer.** Exécute une recherche classique et remplace la fonction selon la valeur fournie pour *find what* et *replace with* (chercher-quoi et remplacer-par). Vous pouvez utiliser des expressions classiques.

  NOTE: Il y a plusieurs implémentations des expressions classiques utilisées. NiceLabel Automation utilise la syntaxe .NET Framework pour les expressions classiques. Pour plus d'informations, consultez la Base de Connaissances article KB250.

- **Remplacer les caractères non-imprimables par un espace.** Remplace tous les caractères de contrôle de la chaîne de caractères par des caractères d'espacement (code décimal ASCII 32). Les caractères non-imprimables sont des caractères à valeurs ASCII entre 0-31 et 127-159.
- **Effacer les caractères non-imprimables.** Efface tous les caractères dans la chaîne de caractères. Les caractères non-imprimables sont des caractères à valeurs ASCII entre 0-31 et 127-159.
- **Décoder les caractères spéciaux.** Les caractères spéciaux (ou caractères de contrôle) sont des caractères qui ne sont pas disponibles sur le clavier, tels que Retour Charriot et Passage à la Ligne. NiceLabel Automation utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Charriot et <LF> pour Passage à la Ligne. Pour plus d'informations, consultez l'article Introduction de Caractères Spéciaux (Codes de Contrôle). Cette option convertit les caractères spéciaux de la syntaxe NiceLabel en caractères binaires réels.

  EXEMPLE: Quand vous recevez les données "<CR><LF>", NiceLabel Automation les utilisera comme une chaîne complète de 8 caractères. Pour interpréter et utiliser les données comme deux caractères binaires `CR` (Retour Charriot - code ASCII 13) et `LF` (Passage à la Ligne - code ASCII 10), vous devrez activer cette nouvelle option.

- **Rechercher et effacer tout avant.** Trouve la chaîne de caractères fournie et efface tous les caractères du début des données jusqu'à la chaîne de caractères. La chaîne de caractères trouvée peut aussi être effacée.
- **Rechercher et effacer tout après.** Trouve la chaîne de caractères fournie et efface tous les caractères depuis la chaîne de caractères jusqu'à la fin des données. La chaîne de caractères trouvée peut aussi être effacée.

Aperçu des données

Cette section fournit un aperçu de la définition de champ. Quand l'élément défini est sélectionné,

l'aperçu va surligner son emplacement dans les données aperçues.

- **Aperçu nom de fichier.** Spécifie le fichier qui contient l'échantillon de données qui sera analysé dans le filtre. Le fichier d'aperçu est copié de la définition du filtre. Si vous changez le nom du fichier d'aperçu, le nouveau nom de fichier sera sauvegardé.
- **Ouvrir.** Sélectionne un autre fichier selon lequel vous voulez exécuter les règles du filtre.
- **Rafraîchir.** Relance les règles du filtre en fonction du contenu du nom du fichier d'aperçu. La section Aperçu de Données sera mise à jour avec le résultat.

Définition Des Éléments Répétables

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Quand vous avez un élément XML qui survient plusieurs fois dans les données XML, cet élément est répétable. Généralement les éléments répétables contiennent les données pour une seule étiquette. Pour indiquer que vous voulez utiliser les données de tous les éléments répétables, pas seulement le premier, vous devez définir l'élément comme **Bloc de Données (bloc de données)** et activer l'option **Élément Répétable**. Quand le filtre contient une définition d'éléments définis comme bloc de donnée / élément répétable, l'action Utilisation d'un Filtre de Données affichera les éléments répétables avec les espaces réservés imbriqués. Toute action imbriquée sous une telle sous-zone s'exécutera seulement pour les blocs de données à ce niveau.

Exemple

L'élément `<item>` est défini comme **Bloc de Données** et **Élément Répétable.** Ceci indique au filtre d'extraire toutes les occurrences de l'élément `<item>`, pas seulement le premier. Dans ce cas, l'élément **<item>** sera défini comme le sous-niveau dans l'action **Utiliser le Filtre de Données**. Vous devez imbriquer les actions "Ouvrir l'Étiquette" et "Imprimer l'Étiquette" sous cet espace réservé de sous-niveau, pour qu'elles soient bouclées autant de fois qu'il y a d'occurrences de l'élément `<item>`. Dans ce cas, trois fois.

```
<?xml version="1.0" encoding="utf-8"?>
<asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
<asx:values>
<NICELABEL_JOB>
<TIMESTAMP>20130221100527.788134</TIMESTAMP>
<USER>PGRI</USER>
<IT_LABEL_DATA>

<item>
<LBL_NAME>goods_receipt.lbl</LBL_NAME>
 <LBL_PRINTER>Production01</LBL_PRINTER>
 <LBL_QUANTITY>1</LBL_QUANTITY>
 <MAKTX>MASS ONE</MAKTX>
 <MATNR>28345</MATNR>
 <MEINS>KG</MEINS>
 <WDATU>19.01.2012</WDATU>
 <QUANTITY>1</QUANTITY>
 <EXIDV>01234567890123456O</EXIDV>
</item>

<item>
<LBL_NAME>goods_receipt.lbl</LBL_NAME>
 <LBL_PRINTER>Production01</LBL_PRINTER>
 <LBL_QUANTITY>1</LBL_QUANTITY>
```

```
<MAKTX>MASS TWO</MAKTX>
<MATNR>28346</MATNR>
<MEINS>KG</MEINS>
<WDATU>11.01.2011</WDATU>
<QUANTITY>1</QUANTITY>
<EXIDV>012345678901234577</EXIDV>
</item>

<item>
<LBL_NAME>goods_receipt.lbl</LBL_NAME>
<LBL_PRINTER>Production01</LBL_PRINTER>
<LBL_QUANTITY>1</LBL_QUANTITY>
<MAKTX>MASS THREE</MAKTX>
<MATNR>27844</MATNR>
<MEINS>KG</MEINS>
<WDATU>07.03.2009</WDATU>
<QUANTITY>1</QUANTITY>
<EXIDV>012345678901234584</EXIDV>
</item>

</IT_LABEL_DATA>
</NICELABEL_JOB>
</asx:values>
</asx:abap>
```

Définition De La Zone D'Affectation XML

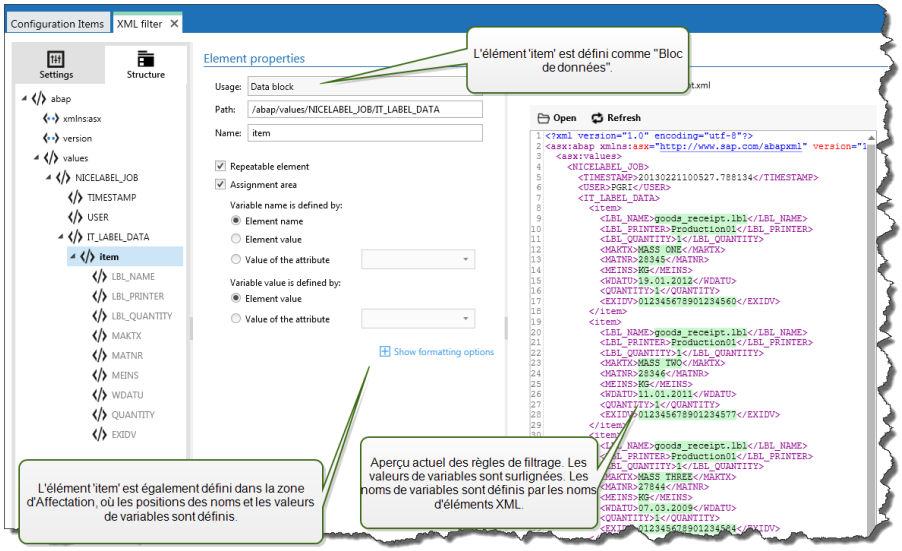
La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Le filtre XML à la possibilité d'identifier automatiquement les champs et leurs valeurs, ce qui élimine la nécessité de mappage manuel *variable à champ*.

Cette fonctionnalité est utile quand le déclencheur reçoit la donnée de la structure changeable. La structure principale de données est identique, par ex. les champs sont délimités par une virgule, ou la même structure XML, mais **l'ordre** dans lequel les champs sont représentés est changé et/ou **le nombre de champs** a changé; il peut y avoir de nouveaux champs, ou certains champs ne sont plus disponibles. La structure sera automatiquement identifiée par le filtre. En même temps, les noms et valeurs de champs (paires `nom:valeur`) seront lues des données, éliminant la nécessité de mapper les variables manuellement.

L'action Utilisation d'un Filtre de Données n'affichera pas de possibilité de mappage, car le mappage se fera de façon dynamique. Vous ne devez même as définir les variables d'étiquettes dans la configuration du déclencheur. L'action assignera les valeurs de champs aux variables d'étiquettes de même nom sans avoir besoin des variables importées de l'étiquette. Toutefois, la règle s'applique seulement à l'action Impression d'étiquette. Si vous voulez utiliser les valeurs de champs dans une autre action, vous devrez définir les variables dans le déclencheur, tout en conservant le mappage automatique *variable-à-champ*.

NOTE: Aucune erreur ne sera signalée si le champ disponible dans les données d'entrée n'a pas de variable d'étiquette correspondant. Les variables manquantes sont ignorées en silence.


Configuration Items
XML filter
Settings
Structure
Element properties
Usage: Data block
Path: /abap/values/NICELABEL_JOB/IT_LABEL_DATA
Name: item
Repeatable element
Assignment area
Variable name is defined by:
Element name
Element value
Value of the attribute
Variable value is defined by:
Element value
Value of the attribute
Show formatting options
Open
Refresh
L'élément 'item' est défini comme "Bloc de données".
L'élément 'item' est également défini dans la zone d'Affectation, où les positions des noms et les valeurs de variables sont définis.
Aperçu actuel des règles de filtrage. Les valeurs de variables sont surlignées. Les noms de variables sont définis par les noms d'éléments XML.

Configuration de la Zone d'Affectation XML

Quand vous définissez le Bloc de Données comme zone d'affectation, deux espaces réservés apparaissent sous la définition de cet élément. Vous devez définir comment le nom de champ et la valeur sont définis pour permettre au filtre d'extraire les paires `nom-valeur`.

- **Nom de variable.** Spécifie l'élément qui contient le nom de champ. Le nom peut être défini par le nom d'élément, l'attribut sélectionné ou la valeur de l'élément. La variable d'étiquette doit avoir le même nom pour permettre le fonctionnement du mappage automatique.
- **Valeur de variable.** Spécifie l'élément qui contient la valeur du champ. Le nom peut être défini par le nom d'élément, l'attribut sélectionné ou la valeur de l'élément.

AVERTISSEMENT L'élément XML qui contient les paires `nom:valeur` ne peuvent pas être dans l'élément racine, mais doit au moins être un élément de second niveau. Par exemple, dans l'élément XML ci-dessous, l'élément `<label>` est l'élément de second niveau et peut contenir les paires `nom:valeur`.

Options de Formatage

Cette section définit les fonctions de manipulation de chaîne de caractères qui seront appliquées aux variables ou champs sélectionnés. Vous pouvez sélectionner une ou plusieurs fonctions. Les fonctions seront appliquées dans l'ordre sélectionné dans l'interface utilisateur, de haut en bas.

- **Enlever les espaces au début.** Enlève tous les caractères d'espacement (code décimal ASCII 32) du début de la chaîne de caractères.
- **Enlever les espaces à la fin.** Enlève tous les caractères d'espacement (code décimal ASCII 32) à la fin de la chaîne de caractères.
- **Enlever les caractères d'ouverture et fermeture.** Efface la première occurrence du caractère d'ouverture et de fermeture trouvé dans la chaîne de caractères.

  EXEMPLE: Si vous utilisez "{" comme caractère d'ouverture et "}" comme caractère de fermeture, la chaîne d'entrée `{{selection}}` sera convertie en `{selection}`.

- **Rechercher et remplacer.** Exécute une recherche classique et remplace la fonction selon la

valeur fournie pour *find what* et *replace with* (chercher-quoi et remplacer-par). Vous pouvez utiliser des expressions classiques.

> **NOTE:** Il y a plusieurs implémentations des expressions classiques utilisées. NiceLabel Automation utilise la syntaxe .NET Framework pour les expressions classiques. Pour plus d'informations, consultez la Base de Connaissances article KB250.

- **Remplacer les caractères non-imprimables par un espace.** Remplace tous les caractères de contrôle de la chaîne de caractères par des caractères d'espacement (code décimal ASCII 32). Les caractères non-imprimables sont des caractères à valeurs ASCII entre 0-31 et 127-159.
- **Effacer les caractères non-imprimables.** Efface tous les caractères dans la chaîne de caractères. Les caractères non-imprimables sont des caractères à valeurs ASCII entre 0-31 et 127-159.
- **Décoder les caractères spéciaux.** Les caractères spéciaux (ou caractères de contrôle) sont des caractères qui ne sont pas disponibles sur le clavier, tels que Retour Charriot et Passage à la Ligne. NiceLabel Automation utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Charriot et <LF> pour Passage à la Ligne. Pour plus d'informations, consultez l'article Introduction de Caractères Spéciaux (Codes de Contrôle). Cette option convertit les caractères spéciaux de la syntaxe NiceLabel en caractères binaires réels.

  EXEMPLE: Quand vous recevez les données "<CR><LF>", NiceLabel Automation les utilisera comme une chaîne complète de 8 caractères. Pour interpréter et utiliser les données comme deux caractères binaires `CR` (Retour Charriot - code ASCII 13) et `LF` (Passage à la Ligne - code ASCII 10), vous devrez activer cette nouvelle option.

- **Rechercher et effacer tout avant.** Trouve la chaîne de caractères fournie et efface tous les caractères du début des données jusqu'à la chaîne de caractères. La chaîne de caractères trouvée peut aussi être effacée.
- **Rechercher et effacer tout après.** Trouve la chaîne de caractères fournie et efface tous les caractères depuis la chaîne de caractères jusqu'à la fin des données. La chaîne de caractères trouvée peut aussi être effacée.

Exemple

L'élément `<label>` est défini comme bloc de données et zone d'affectation. Le **nom de variable** est défini par la valeur du nom d'attribut, **la valeur de variable** est définie par l'élément texte.

```
<?xml version="1.0" standalone="no"?>
<labels _FORMAT="case.lbl" _PRINTERNAME="Production01" _QUANTITY="1">
</label>
<variable name="CASEID">0000000123</variable>
<variable name="CARTONTYPE"/>
<variable name="ORDERKEY">0000000534</variable>
<variable name="BUYERPO"/>
<variable name="ROUTE"> </variable>
<variable name="CONTAINERDETAILID">0000004212</variable>
<variable name="SERIALREFERENCE">0</variable>
<variable name="FILTERVALUE">0</variable>
<variable name="INDICATORDIGIT">0</variable>
<variable name="DATE">11/19/2012 10:59:03</variable>
</label>
</labels>
```

Plus plus d'informations, consultez l'article Exemples.

Paramétrage Des Noms D'Étiquette Et D'Imprimante Des Données D'entrée

Typiquement, les filtres sont utilisés pour extraire les valeurs des données reçues et les envoyer aux variables de l'étiquette pour l'impression. Dans ce cas, le nom de l'étiquette ou de l'imprimante est codé-en-dur dans les actions. Par exemple, l'action Open Label - Ouvrir l'étiquette va coder-en-dur le nom d'étiquette, et l'action Définir l'imprimante va coder-en-dur le nom d'imprimante. Toutefois, les données d'entrée peuvent également fournir les *méta-données*, valeurs utilisées dans le traitement NiceLabel Automation, mais pas imprimées sur l'étiquette, telles que le nom d'étiquette, le nom d'imprimante, la quantité d'étiquettes ou autre chose.

Pour utiliser les valeurs des méta-champs dans le processus d'impression, effectuer les opérations suivantes.

1. **Reconfiguration du filtre.** Vous devez définir des nouveaux champs pour les données d'entrée pour extraire également les champs de méta-données.
2. **Définition de Variable.** Vous devez définir manuellement les variables qui vont sauvegarder les méta-données, elles n'existent pas sur l'étiquette et ne peuvent pas être importées. Utilisez des noms intuitifs, tels que `NomEtiquette`, `NomImprimante`, et `Quantité`. Vous pouvez utiliser n'importe quel nom de variable.
3. **Reconfiguration du mappage.** Vous devez reconfigurer manuellement l'action Utilisation d'un Filtre de Données pour mapper les méta-champs aux variables.
4. **Reconfiguration d'action.** Vous devez reconfigurer l'action **Ouvrir l'Étiquette** pour ouvrir l'étiquette spécifiée par la variable `NomEtiquette`, et l'action "Définir Imprimante" pour utiliser l'imprimante spécifiée par la variable `NomImprimante`.

Exemple

Le fichier CSV contient les données de l'étiquette, mais fournit aussi les *méta-données*, telles que le nom d'étiquette, le nom d'imprimante et la quantité d'étiquettes. Le filtre de Texte Structuré va extraire tous les champs, envoyer les valeurs relatives aux variables d'étiquette et utiliser les *méta-données* pour configurer les actions "Ouvrir Étiquette", "Définir Imprimante" et "Imprimer l'Étiquette".

```
label_name;label_count;printer_name;art_code;art_name;ean13;weight
label1.lbl;1;CAB A3 203DPI;00265012;SAC.PESTO 250G;383860026501;1,1 kg
label2.lbl;1;Zebra R-402;00126502;TAGLIOLINI 250G;383860026002;3,0 kg
```

Plus plus d'informations, consultez l'article Exemples.

Configuration Des Déclencheurs

Comprendre Les Déclencheurs

Les fonctionnalités de cet élément ne sont pas toutes disponibles dans chaque produit NiceLabel Automation.

NiceLabel Automation est une application basée sur des évènements qui va déclencher l'exécution d'actions lors d'un changement de l'évènement surveillé. Vous pouvez utiliser tout déclencheur disponible pour surveiller les changements dans les évènements, tels que la dépose d'un fichier dans un certain dossier, des données arrivant sur un socket TCP/IP spécifique, un message HTTP ou autres. La tâche principale du déclencheur est de reconnaitre le changement dans l'évènement, récupérer les données fournies par l'évènement et ensuite effectuer les actions. La majorité des déclencheurs sont conçus pour écouter passivement l'apparition de l'évènement surveillé, mais il y a deux exceptions. Le **Déclencheur de base de données** est un déclencheur actif qui recherche périodiquement les changements dans la base de données. Le **Déclencheur de port série** qui peut attendre les connexions entrantes, ou peut requérir activement des données à des intervalles de temps déterminés.

Traitement des déclencheurs

Généralement le déclencheur reçoit des données qui doivent être imprimées sur les étiquettes. Dès que le déclencheur reçoit les données, les actions sont exécutées dans l'ordre défini, du haut vers le bas. Les données reçues peuvent contenir des valeurs pour les étiquettes. Toutefois, avant de pouvoir utiliser ces valeurs, vous devez les extraire des données reçues et les sauvegarder dans les variables. Les filtres définissent les règles d'extraction. Quand ils sont exécutés, les filtre sauvegardent les valeurs extraites vers les variables mappées. Dès que les données sont sauvegardées dans les variables, vous pouvez lancer les actions qui vont utiliser les variables, telle que l'impression d'étiquette.

Quand l'évènement apparait, la donnée d'entrée est fournie et sauvegardée dans un fichier temporaire sur le disque, dans le dossier de service de l'utilisateur `%temp`. La variable interne `DataFileName` fait référence à l'emplacement du fichier temporaire. Le fichier est effacé quand le déclencheur termine son exécution.

Caractéristiques du déclencheur

Pour configurer le déclencheur, vous devez définir comment vous allez accepter la donnée et les actions que vous voulez lancer. En option, vous pouvez également utiliser des variables. La configuration du déclencheur comprend trois sections.

- **Paramètres.** Définit les paramètres principaux du déclencheur. Vous pouvez définir l'évènement qui sera surveillé pour activer le déclencheur, ou définir le canal de communication entrante. Les paramètres comprennent la sélection du moteur du script de programmation et les options de sécurité. Les options disponibles dépendent du type de déclencheur. Pour plus d'informations, consultez la section Types de Déclencheurs ci-dessous.
- **Variables** Cette section définit les variables nécessaires dans le déclencheur. Généralement les variables sont importées du modèle d'étiquette; vous pouvez donc les mapper avec les champs extraits des données d'entrée. Vous pouvez aussi définir les variables à utiliser en interne dans les différentes actions et qui ne seront pas envoyées à l'étiquette. Pour plus d'informations, consultez l'article Utilisation de Variables.
- **Actions** Cette section définit les actions à exécuter à chaque fois que le déclencheur détecte un changement dans l'évènement surveillé. Les actions sont exécutées dans l'ordre, de haut en bas. Pour plus d'informations, consultez l'article Utilisation des Actions.

Types de Déclencheurs

- **Définition des Déclencheurs.** Surveille les changements dans un fichier ou un ensemble de fichiers dans le dossier. Le contenu du fichier peut être analysé dans des filtres et utilisé dans des actions.
- **Déclencheur de Port Série.** Surveille la communication entrant sur le port série RS232. Le contenu du flux entrant peut être analysé dans des filtres et utilisé dans des actions. Les données peuvent également être requises à l'appareil externe à des intervalles de temps définis.
- **Déclencheur de Base de Données.** Surveille les changements d'enregistrements dans les tables de la base de données SQL. Le contenu des données retournées peuvent être analysées et utilisées dans les actions. La base de données est surveillée à des intervalles de temps définis. Le déclencheur peut aussi mettre la base de données à jour après avoir exécuté les actions en utilisant les instructions `INSERT`, `UPDATE` et `INSERT SQL`.
- **Déclencheur Serveur TCP/IP.** Surveille le flux de données brutes entrant sur le socket défini. Le contenu du flux entrant peut être analysé dans des filtres et utilisé dans des actions. Il peut être bidirectionnel, fournissant un retour.
- **Déclencheur de Serveur HTTP.** Surveille le flux de données en format HTTP arrivant sur le socket défini. Le contenu du flux entrant peut être analysé dans des filtres et utilisé dans des actions. L'authentification d'utilisateur peut être activée. Il est bidirectionnel, fournissant un feedback.
- **Déclencheur Web Service.** Surveille le flux de données entrant sur la méthode Web Service définie. Le contenu du flux entrant peut être analysé dans des filtres et utilisé dans des actions. Il est bidirectionnel, fournissant un feedback.

Traitement des Erreurs dans les Déclencheurs

- **Erreurs de Configuration.** Le déclencheur sera en état d'erreur tant qu'il n'est pas configuré convenablement ou complètement. Par exemple, vous avez configuré le fichier de déclenchement mais n'avez pas spécifié le nom de fichier à surveiller pour les changements. Ou cous avez défini l'action pour imprimer des étiquettes mais n'avez pas spécifié le nom de l'étiquette. Vous pouvez sauvegarder les déclencheurs qui contiennent des erreurs mais ne pouvez pas les lancer dans le Automation Manager tant que le problème n'est pas résolu. L'erreur dans le niveau inférieur de la configuration se propagera jusqu'au niveau le plus haut,

ce qui rendra la localisation de l'erreur plus facile.

> **EXEMPLE:** Par exemple, si vous avez une action en état d'erreur, toutes les actions de niveau supérieur indiqueront la situation d'erreur, l'icône d'erreur sera affiché dans l'onglet Action et dans le nom du déclencheur.

- **Chevauchements de Configurations.** Il est tout à fait acceptable d'inclure des déclencheurs qui surveillent le même évènement dans la configuration, tels que le même nom de fichier ou l'écoute sur le même port TCP/IP, mais de tels déclencheurs ne peuvent pas fonctionner simultanément. Quand vous démarrer le déclencheur dans Automation Manager, il démarrera seulement si aucun autre déclencheur de la même ou d'une autre configuration ne surveille le même évènement.

Feedback de la tâche d'impression

Pour plus d'informations, consultez l'article Feedback de la tâche d'impression.

Définition Des Déclencheurs

Déclencheur De Fichier

Pour en savoir plus sur les déclencheurs en général, consultez l'article Comprendre les Déclencheurs.

L'évènement d'activation de fichier survient quand un fichier ou un ensemble de fichiers dans un dossier surveillé change. L'apparition d'un nouveau fichier active aussi un activateur. Selon la configuration de l'activateur, le système Windows alerte l'activateur qu'un fichier a changé ou l'activateur conserve lui-même une liste horodatée de la dernière-écriture du fichier et se déclenchera quand le fichier a un nouvel horodatage.

Utilisation typique : Le système utilisé actuellement exécute une transaction qui va générer une activation de fichier dans le dossier partagé. Le contenu de données peut être structuré en format CSV, XML et autres formats, ou il peut être structuré dans un format initial. Dans chaque cas, NiceLabel Automation va lire les données, analyser les valeurs en utilisant des filtres et les imprimer sur les étiquettes. Pour plus de renseignements concernant l'analyse et l'extraction de données, consultez l'article Comprendre les Filtres.

Général

Cette section vous permet de configurer les paramètres des déclencheurs de fichiers les plus importants.

- **Nom.** Spécifie le nom de déclencheur unique. Les noms vous aident à distinguer les différents déclencheurs quand vous les configurez dans Automation Builder et les exécutez ensuite dans Automation Manager.
- **Description.** Procure la possibilité de décrire la fonctionnalité de ce déclencheur. Vous pouvez l'utiliser pour écrire une courte description de la fonction du déclencheur.
- **Détecter le fichier spécifié.** Spécifie le chemin et le nom du fichier que vous allez surveiller pour les changements.

- **Détecte un ensemble de fichiers dans le dossier spécifié.** Spécifie le chemin vers le dossier que vous allez surveiller pour les changements de fichiers et des noms de fichiers. Vous pouvez également utiliser les caractères génériques standards de Windows "*" et "?". Certains types de fichiers sont prédéfinis dans les listes déroulantes mais vous pouvez également introduire vos propres types.

  NOTE: Durant la surveillance de dossiers réseau, assurez-vous d'utiliser la notation UNC de `\\server\share\file`. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

- **Détection automatique de changements.** L'application répondra aux changements du fichier dès que le fichier a été créé. Vous pouvez l'utiliser quand le dossier surveillé est situé sur le disque local.
- **Vérifier les changements dans le dossier à intervalles de (millisecondes).** L'application va scanner le dossier pour des changements de fichiers dans un intervalle de temps défini. Cette méthode de sondage tend à être plus lente que la détection automatique, mais vous devez l'utiliser pour surveiller les dossiers en réseau.

Exécution

Les options dans la section **Accès Fichiers** spécifient comment l'application accède au fichier d'activation.

- **Ouverture exclusive de fichier.** Spécifie l'ouverture du fichier de déclenchement en mode exclusif. Aucune autre application ne peut accéder au fichier en même temps. C'est une sélection par défaut.
- **Ouvrir le fichier avec permissions en lecture seule.** Spécifie l'ouverture du fichier de déclenchement en mode lecture seule.
- **Ouvrir le fichier avec permissions de lecture et écriture.** Spécifie l'ouverture du fichier de déclenchement en mode lecture/écriture.
- **Période de réessayage d'ouverture du fichier.** Spécifie la période de temps durant laquelle NiceLabel Automation va essayer d'ouvrir le fichier de déclenchement. Si l'accès au fichier n'est pas possible au bout de cette période de temps, NiceLabel Automation va rapporter une erreur.

Les options dans la section **Options de Surveillance** spécifient les possibilités de surveillance des fichiers.

- **Vérifier la taille du fichier.** Active la détection de changements, pas seulement l'horodatage, mais aussi la longueur du fichier. Les changements d'horodatage peuvent passer inaperçus, donc le changement de longueur du fichier peut aider à déclencher les actions
- **Ignorer les fichiers de déclenchement vides.** Si le fichier de déclenchement est vide, il sera ignoré. Les actions ne seront pas exécutées.
- **Effacer le fichier de déclenchement.** Après que le changement a été détecté dans le fichier de déclenchement et le déclencheur est activé, le fichier sera effacé. L'activation de cette option permet de nettoyer le dossier des fichiers déjà traités.

NOTE: NiceLabel Automation crée toujours une sauvegarde de la donnée de déclenchement reçue; dans ce cas, le contenu des fichiers de déclenchement et la sauvegarde dans un fichier à nom unique. C'est important quand vous avez besoin du contenu du fichier de déclenchement pour certaines actions, telle que **Fichier Commande de Lancement**. L'emplacement des données de déclenchement sauvegardées est référencée par la variable interne *DataFileName*.

- **Effacer le contenu du fichier.** L'exécution des actions vide le contenu du fichier de déclenchement. C'est utile quand des applications tierces ajoutent des données au fichier de déclenchement. Vous voulez conserver le fichier pour que l'ajout puisse se faire, mais vous ne voulez pas imprimer des anciennes données.
- **Suivre les changements quand le déclencheur est inactif.** Spécifie si vous voulez activer le déclencheur si les fichiers changent quand le déclencheur n'a pas été démarré. Si votre NiceLabel Automation n'est pas déployé dans l'environnement de haute-disponibilité avec des serveurs de sauvegarde, les fichiers de déclenchement entrants peuvent se perdre quand le serveur est arrêté. Quand le NiceLabel Automation est à nouveau en ligne, les fichiers de déclenchement existants peuvent être traités.

Autre

Les options dans la section **Feedback du Moteur d'Impression** spécifient la communication avec le moteur d'impression.

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

- **Impression surveillée.** Active le mode d'impression synchrone. Utilisez cette option quand vous voulez renvoyer l'état de la tâche d'impression à l'application tierce. Pour plus d'informations, consultez l'article Mode d'impression Synchrone.

Les options dans la section **Traitement de Données** spécifient si vous voulez couper les données pour qu'elles entrent dans la variable ou ignorer les variables d'étiquettes manquantes. Par défaut, NiceLabel Automation rapporte les erreurs et interrompt le processus d'impression si vous essayez de sauvegarder une valeur trop longue dans la variable d'étiquette, ou essayez de paramétrer une valeur dans une variable d'étiquette inexistante.

- **Ignorer les contenus de variable excessifs.** Les valeurs de données dépassant la longueur de variable définie dans le concepteur d'étiquettes seront tronquées pour pouvoir entrer dans la variable. Cette option s'applique lors du paramétrage de valeurs dans les filtres des fichiers de commande et au paramétrage de valeurs de variables de déclencheurs dans les variables d'étiquette de même nom.

  EXEMPLE: La variable d'étiquette accepte un maximum de 10 caractères. Quand cette option est activée, toute valeur de plus de 10 caractères sera tronquée au 10 premiers caractères, tous les caractères suivant le caractère numéro 10 seront ignorés.

- **Ignorer les variables d'étiquettes manquantes.** Quand vous effectuez l'impression avec des fichiers de commande (tel qu'un fichier JOB), le processus d'impression va ignorer toutes les variables qui sont spécifiées dans le fichier de commande (utilisant la commande SET), mais ne sont pas définies dans l'étiquette. Il n'y aura pas d'erreur lors de l'essai de paramétrage de la

variable d'étiquette inexistante. Un processus semblable s'effectue quand vous définissez une zone d'affectation dans le filtre pour extraire toutes les paires *nom:valeur*, mais vous avez moins de variables définies dans l'étiquette.

Les options dans la section **Script** spécifient les possibilités de script.

- **Langage de Script.** Spécifie le langage de script activé pour le déclencheur. Toutes les actions **Execute script** que vous utilisez dans un même déclencheur utilisent le même langage.

Les options dans la section **Sauvegarder les Données Reçues** spécifient les commandes disponibles pour les données reçues par le déclencheur.

- **Sauvegarder les données reçues par le déclencheur dans un fichier.** Activez cette option pour sauvegarder les données reçues par le déclencheur. L'option **Variable** active le nom de fichier variable. Vous devez sélectionner une variable qui contient le chemin et le nom du fichier.
- **En cas d'erreur, sauvegarder les données reçues par le déclencheur dans un fichier.** Activez cette option pour sauvegarder les données dans le déclencheur uniquement si l'erreur se produit durant l'exécution de l'action. Vous allez activez cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème ultérieurement.

NOTE: NiceLabel Automation sauvegarde toujours les données reçues dans un nom de fichier temporaire, qui est effacé dès la fin d'exécution du déclencheur . La variable interne `DataFileName` pointe vers ce nom de fichier. Pour plus d'informations, consultez l'article Variables Internes.

Sécurité

- **Verrouiller et encoder le déclencheur.** Active la protection du déclencheur. Quand elle est activée, le déclencheur est verrouillé et ne peut pas être édité; les actions deviennent encodées. Seul l'utilisateur ayant le mot de passe peut déverrouiller le déclencheur et le modifier.

Déclencheur De Port Série

Pour en savoir plus sur les déclencheurs en général, consultez l'article Comprendre les Déclencheurs.

L'évènement du déclencheur de port série s'active quand des données sont reçues sur le port série RS232 surveillé.

Utilisation typique : **(1) Remplacement d'imprimante.** Vous allez mettre hors service l'imprimante existante d'étiquettes connectée sur le port série. NiceLabel Automation va accepter les données à sa place, extraire les valeurs pour les objets d'étiquettes du flux d'impression reçu et créer les tâches d'impression pour le nouveau modèle d'imprimante. **(2) Balances.** Les balances procurent les données concernant les objets pesés. NiceLabel Automation extrait les données requises du flux de données reçu et imprime une étiquette. Pour plus de renseignements concernant l'analyse et l'extraction de données, consultez l'article Comprendre les Filtres.

Général

Cette section vous permet de configurer les paramètres des déclencheurs de fichiers les plus importants.

- **Nom.** Spécifie le nom de déclencheur unique. Les noms vous aident à distinguer les différents déclencheurs quand vous les configurez dans Automation Builder et les exécutez ensuite dans Automation Manager.
- **Description.** Procure la possibilité de décrire la fonctionnalité de ce déclencheur. Vous pouvez l'utiliser pour écrire une courte description de la fonction du déclencheur.
- **Port.** Spécifie le numéro de port série (COM) sur lequel les données entrantes seront acceptées. Utilisez le port qui n'est pas utilisé par une autre application, ou appareil, tel que le pilote d'imprimante. Si le port sélectionné est utilisé, vous ne pourrez pas démarrer le déclencheur dans le Automation Manager.

Les options dans la section **Paramétrage du Port** spécifient les paramètres de communication qui doivent correspondre aux paramètres assignés à l'appareil à port série.

- **Désactiver l'initialisation du port.** Spécifie que l'initialisation du port ne sera pas effectuée quand vous lancez le déclencheur dans le Automation Manager. Cette option est parfois requise pour les ports COM virtuels.

Exécution

- **Utiliser les données d'initialisation.** Spécifie que vous voulez envoyer la chaîne d'initialisation à l'appareil série chaque fois que le déclencheur est activé. Certains appareils série doivent être réveillés ou mis en mode de veille avant qu'ils ne puissent procurer les données. Pour plus d'informations concernant la chaîne d'initialisation et si vous avez besoin de tout, veuillez consulter le guide d'utilisation. Vous pouvez inclure des caractères binaires. Pour plus d'informations, consultez l'article Introduction de Caractères Spéciaux (Codes de Contrôle).
- **Utiliser le sondage de données.** Spécifie que le déclencheur demandera les donnée de l'appareil de façon active. Le déclencheur va envoyer les commandes fournies dans les champs de Contenu aux intervalles réguliers spécifiés. peut inclure des caractères binaires. Pour plus d'informations, consultez l'article Introduction de Caractères Spéciaux (Codes de Contrôle).

Autre

Les options dans la section **Feedback du Moteur d'Impression** spécifient la communication avec le moteur d'impression.

> La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

- **Impression surveillée.** Active le mode d'impression synchrone. Utilisez cette option quand vous voulez renvoyer l'état de la tâche d'impression à l'application tierce. Pour plus d'informations, consultez l'article Mode d'impression Synchrone.

Les options dans la section **Traitement de Données** spécifient si vous voulez couper les données pour qu'elles entrent dans la variable ou ignorer les variables d'étiquettes manquantes. Par défaut, NiceLabel Automation rapporte les erreurs et interrompt le processus d'impression si vous essayez de sauvegarder une valeur trop longue dans la variable d'étiquette, ou essayez de paramétrer une valeur dans une variable d'étiquette inexistante.

- **Ignorer les contenus de variable excessifs.** Les valeurs de données dépassant la longueur de variable définie dans le concepteur d'étiquettes seront tronquées pour pouvoir entrer dans la variable. Cette option s'applique lors du paramétrage de valeurs dans les filtres des fichiers de

commande et au paramétrage de valeurs de variables de déclencheurs dans les variables d'étiquette de même nom.

> **EXEMPLE:** La variable d'étiquette accepte un maximum de 10 caractères. Quand cette option est activée, toute valeur de plus de 10 caractères sera tronquée au 10 premiers caractères, tous les caractères suivant le caractère numéro 10 seront ignorés.

- **Ignorer les variables d'étiquettes manquantes.** Quand vous effectuez l'impression avec des fichiers de commande (tel qu'un fichier JOB), le processus d'impression va ignorer toutes les variables qui sont spécifiées dans le fichier de commande (utilisant la commande SET), mais ne sont pas définies dans l'étiquette. Il n'y aura pas d'erreur lors de l'essai de paramétrage de la variable d'étiquette inexistante. Un processus semblable s'effectue quand vous définissez une zone d'affectation dans le filtre pour extraire toutes les paires *nom:valeur*, mais vous avez moins de variables définies dans l'étiquette.

Les options dans la section **Script** spécifient les possibilités de script.

- **Langage de Script.** Spécifie le langage de script activé pour le déclencheur. Toutes les actions **Execute script** que vous utilisez dans un même déclencheur utilisent le même langage.

Les options dans la section **Sauvegarder les Données Reçues** spécifient les commandes disponibles pour les données reçues par le déclencheur.

- **Sauvegarder les données reçues par le déclencheur dans un fichier.** Activez cette option pour sauvegarder les données reçues par le déclencheur. L'option **Variable** active le nom de fichier variable. Vous devez sélectionner une variable qui contient le chemin et le nom du fichier.
- **En cas d'erreur, sauvegarder les données reçues par le déclencheur dans un fichier.** Activez cette option pour sauvegarder les données dans le déclencheur uniquement si l'erreur se produit durant l'exécution de l'action. Vous allez activez cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème ultérieurement.

> **NOTE:** NiceLabel Automation sauvegarde toujours les données reçues dans un nom de fichier temporaire, qui est effacé dès la fin d'exécution du déclencheur . La variable interne `DataFileName` pointe vers ce nom de fichier. Pour plus d'informations, consultez l'article Variables Internes.

Sécurité

- **Verrouiller et encoder le déclencheur.** Active la protection du déclencheur. Quand elle est activée, le déclencheur est verrouillé et ne peut pas être édité; les actions deviennent encodées. Seul l'utilisateur ayant le mot de passe peut déverrouiller le déclencheur et le modifier.

Déclencheur De Base De Données

Pour en savoir plus sur les déclencheurs en général, consultez l'article Comprendre les Déclencheurs.

L'évènement de déclenchement de la base de données survient quand un changement est détecté dans la base de données surveillée. Il peut s'agir de nouvelles données ou de données existantes qui ont été mises à jour. Le déclencheur de la base de données n'attend pas un évènement de changement, tel que l'arrivée de données. En fait, il extrait les données de la base de données à des intervalles de temps définis.

Utilisation typique : Le système d'application existant exécute une transaction qui a pour effet de mettre à jour certaines données dans une table de la base de données. NiceLabel Automation va détecter les données mises à jour et les nouvelles données et va imprimer leurs contenus sur les étiquettes.

Général

Cette section vous permet de configurer les paramètres des déclencheurs de fichiers les plus importants.

- **Nom.** Spécifie le nom de déclencheur unique. Les noms vous aident à distinguer les différents déclencheurs quand vous les configurez dans Automation Builder et les exécutez ensuite dans Automation Manager.
- **Description.** Procure la possibilité de décrire la fonctionnalité de ce déclencheur. Vous pouvez l'utiliser pour écrire une courte description de la fonction du déclencheur.
- **Connexion à la base de données.** Spécifie le lien de connexion à la base de données. L'interface de la base de données s'ouvre en cliquant sur le bouton **Définir**, dans lequel vous pouvez configurer une connexion à la base de données, y compris le type de base de données, le nom de la table et les détails de l'utilisateur. Vous devez vous connecter à la base de données permettant l'accès avec des commandes SQL. Pour cette raison, vous ne pouvez pas utiliser le déclencheur de base de données pour détecter les changements de données dans les fichiers texte CSV (fichiers séparés par virgule) et feuilles de calcul Microsoft Excel.

  NOTE: Les détails de configuration dépendent du type de base de données sélectionné. Les option dans l'interface de dialogue dépendent des pilotes de base de donnée que vous utilisez. Pour les détails de configuration, consultez le guide de votre pilote de base de données. Pour plus d'informations concernant la connectivité de la base de données, consultez l'article Accéder aux bases de données.

- **Vérification de la base de données à intervalles de temps.** Spécifie l'intervalle de temps pour sonder les enregistrements de la base de données.
- **Options de détection et Avancées.** Ces options permettent d'affiner le mécanisme de détection d'enregistrements. Quand les enregistrements sont obtenus de la base de données, l'onglet Action affichera automatiquement l'objet pour Chaque Enregistrement, où vous pouvez mapper les champs des tables aux variables des étiquettes.

  Obtention d'enregistrements basée sur une valeur incrémentale unique de champ

  Dans ce cas, le déclencheur surveillera un champ spécifique auto-incrémental numérique de la table. NiceLabel Automation mémorise la valeur du champ pour le dernier enregistrement. A l'intervalle de sondage suivant, seules les valeurs plus grandes que celles mémorisées seront acquises. Pour configurer cette option, vous devez sélectionner le nom de la table où les données résident `(nom de table)`, le champ auto-incrémental `(champ clé)` et la valeur de départ du champ `(valeur du champ clé)`. En interne, la variable `KeyField` est utilisée pour se référencer à la valeur actuelle du champ clé.

NOTE: La dernière valeur du champ clé est mémorisée en interne, mais n'est pas remise à jour dans la configuration, donc la valeur pour `valeur par défaut du champ clé` ne change pas dans cette fenêtre de dialogue. Vous pouvez recharger la configuration en toute sécurité et/ou démarrer/arrêter ce déclencheur dans le Automation Manager tout en conservant la dernière valeur mémorisée. Toutefois, si vous enlevez la configuration du Automation Manager et la remettez, la dernière valeur mémorisée du champ clé sera remise à la valeur définie dans la `valeur par défaut du champ clé`.

Obtenir les enregistrements et les effacer

Dans ce cas, tous les enregistrements sont acquis de la table et ensuite effacés de la table. Pour configurer cette option, vous devez sélectionner le nom de la table où résident les enregistrements (`nom de table`) et spécifier la clé primaire dans la table (`champs de clés`). Il peut arriver d'avoir un tableau sans clé primaire, il est toutefois fortement recommandé de définir une clé primaire. Si la clé primaire existe, les données seront effacées une par une quand la donnée particulière est traitée dans les actions.

AVERTISSEMENT Si la clé primaire n'existe pas, toutes les données obtenues dans le déclencheur actuel seront effacées en même temps. C'est bien quand il n'y a pas d'erreurs de traitement des données. Mais si une erreur survient durant le traitement des données, l'Automation va arrêter le traitement de toutes les données. Comme toutes les données capturés durant cet intervalle de sondage ont déjà été effacées sans avoir été traitées, cela peut causer une perte de données. C'est pourquoi c'est une bonne idée d'avoir une clé primaire.

Exemples de Code SQL Code

NOTE: Ces instructions SQL sont en lecture-seule et sont seulement fournis comme référence. Pour fournir les instructions SQL personnalisée, sélectionnez la méthode de détection **Obtenir et gérer les données par une SQL personnalisée**.

Table d'exemple.

ID	ProductID	CodeEAN	ProductDesc	AlreadyPrinted
1	CAS0006	8021228110014	CASONCELLI ALLA CARNE 250G	Y
2	PAS501	8021228310001	BIGOLI 250G	
3	PAS502GI	8021228310018	TAGLIATELLE 250G	

Exemple d'instruction Update SQL (mise à jour), quand la table contient l'index primaire.

```
DELETE FROM [Table]
WHERE [ID] = :ID
```

Le champ `ID` dans la table est défini comme l'indexe primaire. La construction `:ID` dans la clause WHERE contient la valeur du champ ID dans chaque itération. Pour la première donnée, la valeur de `ID` est 1, pour la deuxième donnée 2, etc. L'usage de la variable est spécifié par le double point devant le nom de champ dans l'instruction SQL.

Exemple d'instruction SQL Update, quand la table n'a pas d'indexe primaire défini.

```
DELETE FROM [Table]
```

Quand l'indexe primaire n'est pas défini dans la table, toutes les données seront effacées de la table quand la première donnée a été traitée.

Obtenir les enregistrements et les mettre à jour

Dans ce cas, tous les enregistrements sont acquis de la table et ensuite mis à jour. Vous pouvez inscrire une valeur personnalisée dans un champ de la table comme indication que 'ces données ont déjà été imprimées'. Pour configurer cette option, vous devez sélectionner le nom de la table où résident les données `(nom table)`, sélectionner le champ que vous voulez mettre à jour `(champ mis à jour)` et introduire la valeur qui sera enregistrée dans le champ `(valeur mise à jour)`. En interne, la variable `UpdateValue` est utilisée dans l'instruction SQL pour référencer la valeur actuelle du champ (`update value`) - mise à jour de valeur.

Il peut arriver d'avoir un tableau sans clé primaire, il est toutefois fortement recommandé de définir une clé primaire. Si la clé primaire existe, les données seront mises à jour une par une quand la donnée particulière est traitée dans les actions.

AVERTISSEMENT Si la clé primaire n'existe pas, toutes les données obtenues dans le déclencheur actuel seront mises à jour en même temps. C'est bien quand il n'y a pas d'erreurs de traitement des données. Mais si une erreur survient durant le traitement des données, l'Automation va arrêter le traitement de toutes les données. Car toutes les données capturés durant cet intervalle de sondage ont déjà été mises à jour sans avoir été traitées dans les actions. C'est pourquoi c'est une bonne idée d'avoir une clé primaire.

Exemples de Code SQL

NOTE: Ces instructions SQL sont en lecture-seule et sont seulement fournis comme référence. Pour fournir les instructions SQL personnalisée, sélectionnez la méthode de détection **Obtenir et gérer les données par une SQL personnalisée**.

Table d'exemple.

ID	ProductID	CodeEAN	ProductDesc	AlreadyPrinted
1	CAS0006	8021228110014	CASONCELLI ALLA CARNE 250G	Y
2	PAS501	8021228310001	BIGOLI 250G	
3	PAS502GI	8021228310018	TAGLIATELLE 250G	

Exemple d'instruction Update SQL (mise à jour), quand la table contient l'index primaire.

```
UPDATE [Table]
SET [AlreadyPrinted] = :UpdateValue
WHERE [ID] = :ID
```

Le champ `ID` dans la table est défini comme l'indexe primaire. La construction `:ID` dans la clause WHERE contient la valeur du champ ID dans chaque itération. Pour la première donnée, la valeur de `ID` est 1, pour la deuxième donnée 2, etc. L'usage de la variable est spécifié par le

double point devant le nom de champ dans l'instruction SQL. Le champ `UpdateValue` est défini dans la configuration de déclencheur, dans le champ d'édition **Update value**.

Exemple d'instruction SQL Update, quand la table n'a pas d'indexe primaire défini.

```
UPDATE [Table]
SET [AlreadyPrinted] = :UpdateValue
```

Quand l'indexe primaire n'est pas défini dans la table, toutes les données de la table seront mises à jour quand la première donnée a été traitée.

Obtenir et gérer les données avec SQL personnalisée

Dans ce cas, l'instruction SQL pour l'extraction de donnée et les mises à jour des champs sont entièrement à votre charge. Pour confirmer cette option, vous devez introduire une instruction SQL personnalisée pour acquérir les données `(instruction de recherche SQL)` et une instruction SQL personnalisée pour mettre à jour l'enregistrement après le traitement `(instruction SQL mise à jour)`. Cliquez sur le bouton **Test** pour exécuter votre instruction SQL et visualiser le résultat à l'écran.

Vous pouvez utiliser les champs de la table ou les valeurs de variables du déclencheur comme paramètres dans la clause WHERE de l'instruction SQL. Vous utiliserez le caractère double-point (:) devant le nom de champ ou de variable. Ceci signale au NiceLabel Automation d'utiliser la valeur actuelle de ce champ ou variable.

Exemples de Code SQL

Table d'exemple.

ID	ProductID	CodeEAN	ProductDesc	AlreadyPrinted
1	CAS0006	8021228110014	CASONCELLI ALLA CARNE 250G	Y
2	PAS501	8021228310001	BIGOLI 250G	
3	PAS502GI	8021228310018	TAGLIATELLE 250G	

Exemple d'instruction SQL de Recherche (Search).

Pour obtenir les données qui n'ont pas encore été imprimées, effectuez les opérations suivantes. Le champ `AlreadyPrinted` (déjà imprimé) ne peut pas contenir la valeur `Y`, et avoir une valeur vide ou NULLE.

```
SELECT * FROM Table
WHERE AlreadyPrinted <> 'Y' or AlreadyPrinted is NULL
```

Deux données vont être extraites de la table ci-dessus, avec les valeurs ID 2 et 3. La première donnée a déjà été imprimée et sera ignorée.

Exemple d'instruction SQL mise à jour (update).

Pour marquer les données déjà imprimées par un `Y` dans le champ `AlreadyPrinted` (déjà imprimé), effectuez les opérations suivantes.

```
UPDATE [Table]
SET [AlreadyPrinted] = 'Y'
WHERE [ID] = :ID
```

Vous devez mettre le double-point (`:`) devant le nom de variable dans votre instruction SQL pour l'identifier comme une variable. Vous pouvez utiliser n'importe quel champ de la table pour les paramètres de la clause WHERE. Dans l'exemple, nous mettons à jour le champ `AlreadyPrinted` (déjà imprimé) seulement pour les données traitées actuellement (la valeur du champ `ID` doit être la même que la valeur de la donnée actuelle). De façon similaire, vous allez référencer les autres champs dans les données comme `:ProductID` ou `:CodeEAN`, ou même référencer les variables définies dans le déclencheur de base de données.

Pour effacer la donnée actuelle de la table, effectuez les opérations suivantes.

```
DELETE FROM [Table]
WHERE [ID] = :ID
```

Afficher l'instruction SQL. Élargissez cette section pour afficher l'instruction SQL générée et écrire votre propre instruction, si vous avez sélectionné l'option **Obtenir et gérer les enregistrements avec SQL personnalisée**.

Aperçu d'exécution SQL

Pour tester l'exécution des instructions SQL et visualiser les effets, cliquez sur le bouton Test dans la barre d'outils de la zone d'édition SQL. La section d'aperçu SQL s'ouvre dans la panneau de droite. Cliquez sur le bouton **Exécuter** pour démarrer le code SQL. Quand vous utilisez les valeurs de champs du tableau (avec le signe (:) devant le nom de champ) dans l'instruction SQL, vous devrez leur fournir les valeurs de test.

NOTE: Si vous avez l'Aperçu des Données ouvert et venez d'ajouter des variables au script, cliquez le bouton **Test** deux fois (pour fermer et ouvrir la section Aperçu de Données) pour mettre à jour la liste de variables dans l'aperçu.

- **Simulation d'exécution.** Spécifie que tous les changements effectués dans la base de données sont ignorés. Les transactions dans la base de données sont annulées, donc aucune mise à jour n'est inscrite dans la base de données.

Exécution

Les options dans Exécution spécifient quand les mises à jour de la base de données s'effectuent. Le type de mise à jour dépend des Options de Détection pour l'activateur.

- **Avant l'exécution des actions.** Spécifie que les enregistrements seront mis à jour avant que les actions définies pour cet activateur sont exécutées.
- **Après l'exécution des actions.** Spécifie que les enregistrements seront mis à jour après que les actions définies pour cet activateur sont exécutées. Généralement vous voulez mettre les données à jour après qu'elles ont été traitées avec succès.

NOTE: Si nécessaire, vous pouvez aussi mettre les données à jour durant l'exécution des actions. Pour plus d'informations, voir l'article Exécuter une instruction SQL.

Autre

Les options dans la section **Feedback du Moteur d'Impression** spécifient la communication avec le moteur d'impression.

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

- **Impression surveillée.** Active le mode d'impression synchrone. Utilisez cette option quand vous voulez renvoyer l'état de la tâche d'impression à l'application tierce. Pour plus d'informations, consultez l'article Mode d'impression Synchrone.

Les options dans la section **Traitement de Données** spécifient si vous voulez couper les données pour qu'elles entrent dans la variable ou ignorer les variables d'étiquettes manquantes. Par défaut, NiceLabel Automation rapporte les erreurs et interrompt le processus d'impression si vous essayez de sauvegarder une valeur trop longue dans la variable d'étiquette, ou essayez de paramétrer une valeur dans une variable d'étiquette inexistante.

- **Ignorer les contenus de variable excessifs.** Les valeurs de données dépassant la longueur de variable définie dans le concepteur d'étiquettes seront tronquées pour pouvoir entrer dans la variable. Cette option s'applique lors du paramétrage de valeurs dans les filtres des fichiers de commande et au paramétrage de valeurs de variables de déclencheurs dans les variables d'étiquette de même nom.

  EXEMPLE: La variable d'étiquette accepte un maximum de 10 caractères. Quand cette option est activée, toute valeur de plus de 10 caractères sera tronquée au 10 premiers caractères, tous les caractères suivant le caractère numéro 10 seront ignorés.

- **Ignorer les variables d'étiquettes manquantes.** Quand vous effectuez l'impression avec des fichiers de commande (tel qu'un fichier JOB), le processus d'impression va ignorer toutes les variables qui sont spécifiées dans le fichier de commande (utilisant la commande SET), mais ne sont pas définies dans l'étiquette. Il n'y aura pas d'erreur lors de l'essai de paramétrage de la variable d'étiquette inexistante. Un processus semblable s'effectue quand vous définissez une zone d'affectation dans le filtre pour extraire toutes les paires *nom:valeur*, mais vous avez moins de variables définies dans l'étiquette.

Les options dans la section **Script** spécifient les possibilités de script.

- **Langage de Script.** Spécifie le langage de script activé pour le déclencheur. Toutes les actions **Execute script** que vous utilisez dans un même déclencheur utilisent le même langage.

Les options dans la section **Sauvegarder les Données Reçues** spécifient les commandes disponibles pour les données reçues par le déclencheur.

- **Sauvegarder les données reçues par le déclencheur dans un fichier.** Activez cette option pour sauvegarder les données reçues par le déclencheur. L'option **Variable** active le nom de fichier variable. Vous devez sélectionner une variable qui contient le chemin et le nom du fichier.
- **En cas d'erreur, sauvegarder les données reçues par le déclencheur dans un fichier.** Activez cette option pour sauvegarder les données dans le déclencheur uniquement si l'erreur se produit durant l'exécution de l'action. Vous allez activez cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème ultérieurement.

NOTE: NiceLabel Automation sauvegarde toujours les données reçues dans un nom de fichier temporaire, qui est effacé dès la fin d'exécution du déclencheur . La variable interne `DataFileName` pointe vers ce nom de fichier. Pour plus d'informations, consultez l'article Variables Internes.

Sécurité

- **Verrouiller et encoder le déclencheur.** Active la protection du déclencheur. Quand elle est activée, le déclencheur est verrouillé et ne peut pas être édité; les actions deviennent encodées. Seul l'utilisateur ayant le mot de passe peut déverrouiller le déclencheur et le modifier.

Déclencheur Serveur TCP/IP

Pour en savoir plus sur les déclencheurs en général, consultez l'article Comprendre les Déclencheurs.

L'évènement d'activation TCP/IP survient quand les données sont reçues sur le socket surveillé (numéro d'adresse IP et de port).

Utilisation typique : Le système existant exécute une transaction, qui en fait envoie les données au serveur NiceLabel Automation sur un socket spécifique. Le contenu de données peut être structuré en format CSV, XML et autres formats, ou il peut être structuré dans un format initial. Dans chaque cas, NiceLabel Automation va lire les données, analyser les valeurs en utilisant des filtres et les imprimer sur les étiquettes. Pour plus de renseignements concernant l'analyse et l'extraction de données, consultez l'article Comprendre les Filtres.

Général

NOTE: Ce déclencheur support le Protocole Internet version 6 (IPv6).

Cette section vous permet de configurer les paramètres des déclencheurs de fichiers les plus importants.

- **Nom.** Spécifie le nom de déclencheur unique. Les noms vous aident à distinguer les différents déclencheurs quand vous les configurez dans Automation Builder et les exécutez ensuite dans Automation Manager.
- **Description.** Procure la possibilité de décrire la fonctionnalité de ce déclencheur. Vous pouvez l'utiliser pour écrire une courte description de la fonction du déclencheur.
- **Port.** Spécifie le numéro de port sur lequel les données entrantes seront acceptées. Utilisez un numéro de port qui n'est pas utilisé par une autre application. Si le port sélectionné est utilisé, vous ne pourrez pas démarrer le déclencheur dans le Automation Manager. Pour plus d'informations concernant la sécurité, consultez l'article Sécuriser l'accès à vos déclencheurs.

  NOTE: Si votre serveur a l'hébergement-multiple activé (plusieurs adresses IP sur une ou plusieurs cartes réseau, NiceLabel Automation répondra au port défini pour toutes les adresses IP.

- **Nombre maximum de connexions concurrentes.** Spécifie le nombre maximum de connexions acceptées. Autant de clients peuvent envoyer des données au déclencheur.

Les options dans la section **Évènement d'Exécution** spécifient quand le déclencheur doit activer et démarrer l'exécution des actions.

- **A la déconnexion du client.** Spécifie que le déclencheur va s'activer après que le client envoi les données et termine la connexion. C'est un paramètre par défaut.

NOTE: N'utilisez pas cette option si vous voulez renvoyer l'état de la tâche d'impression à l'application tierce comme feedback. Si la connexion reste ouverte, vous pouvez envoyer un feedback en utilisant l'action **Envoyer les données au port TCP/IP** avec le paramètre *Répondre à l'expéditeur*.

- **Selon le nombre de caractères reçus.** Spécifie que le déclencheur va s'activer quand le nombre de caractères requis a été reçu. Dans ce cas l'application de partie tierce peut garder la connexion ouverte et envoyer continuellement des données. Chaque segment de données doit avoir la même taille.
- **Selon la séquence de caractères reçue.** Spécifie que le déclencheur va s'activer chaque fois que le la séquence de caractères requise a été reçue. Vous utiliserez cette option si vous savez que la 'fin de données' est toujours identifiée par une chaîne de caractères unique. Vous pouvez insérer des caractères spéciaux (binaires) en utilisant le bouton à côté du champ d'édition.
    - **Inclure dans les données du déclencheur.** La séquence de caractères qui est utilisée pour déterminer l'évènement de déclenchement ne sera pas coupée des données, mais sera inclue dans les données. Le déclencheur recevra le flux de données complet.
- **Quand rien n'est reçu au bout de l'intervalle de temps spécifié.** Spécifie que le déclencheur va s'activer après un intervalle de temps requis suivant la réception du dernier caractère.

Exécution

- **Permettre les connexions depuis les hôtes suivants.** Spécifie la liste d'adresses IP ou les noms d'hôte des ordinateurs pouvant se connecter au déclencheur. Introduisez chaque entrée sur une nouvelle ligne.
- **Refuser les connexions depuis les hôtes suivants.** Spécifie la liste d'adresses IP ou les noms d'hôte des ordinateurs qui ne peuvent pas se connecter au déclencheur. Introduisez chaque entrée sur une nouvelle ligne.
- **Message de bienvenue.** Spécifie le texte du message qui est renvoyé au client chaque fois qu'il se connecte au déclencheur TCP/IP.
- **Message de réponse.** Spécifie le texte du message qui est renvoyé au client chaque fois que l'action est exécutée. Utilisez cette option quand le client ne se déconnecte pas après la transmission de données et attend la réponse de fin d'exécution de l'action. Le message répondu est codé en dur et est donc toujours le même.
- **Encodage du message.** Spécifie le modèle d'encodage des données, de façon à ce que les caractères spéciaux puissent être traités correctement. NiceLabel Automation peut détecter automatiquement l'encodage des données, basé sur l'entête BOM (fichiers texte), ou l'attribut d'encodage (fichiers XML).

Autre

Les options dans la section **Feedback du Moteur d'Impression** spécifient la communication avec le moteur d'impression.

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

- **Impression surveillée.** Active le mode d'impression synchrone. Utilisez cette option quand vous voulez renvoyer l'état de la tâche d'impression à l'application tierce. Pour plus d'informations, consultez l'article Mode d'impression Synchrone.

Les options dans la section **Traitement de Données** spécifient si vous voulez couper les données pour qu'elles entrent dans la variable ou ignorer les variables d'étiquettes manquantes. Par défaut, NiceLabel Automation rapporte les erreurs et interrompt le processus d'impression si vous essayez de sauvegarder une valeur trop longue dans la variable d'étiquette, ou essayez de paramétrer une valeur dans une variable d'étiquette inexistante.

- **Ignorer les contenus de variable excessifs.** Les valeurs de données dépassant la longueur de variable définie dans le concepteur d'étiquettes seront tronquées pour pouvoir entrer dans la variable. Cette option s'applique lors du paramétrage de valeurs dans les filtres des fichiers de commande et au paramétrage de valeurs de variables de déclencheurs dans les variables d'étiquette de même nom.

  EXEMPLE: La variable d'étiquette accepte un maximum de 10 caractères. Quand cette option est activée, toute valeur de plus de 10 caractères sera tronquée au 10 premiers caractères, tous les caractères suivant le caractère numéro 10 seront ignorés.

- **Ignorer les variables d'étiquettes manquantes.** Quand vous effectuez l'impression avec des fichiers de commande (tel qu'un fichier JOB), le processus d'impression va ignorer toutes les variables qui sont spécifiées dans le fichier de commande (utilisant la commande SET), mais ne sont pas définies dans l'étiquette. Il n'y aura pas d'erreur lors de l'essai de paramétrage de la variable d'étiquette inexistante. Un processus semblable s'effectue quand vous définissez une zone d'affectation dans le filtre pour extraire toutes les paires *nom:valeur*, mais vous avez moins de variables définies dans l'étiquette.

Les options dans la section **Script** spécifient les possibilités de script.

- **Langage de Script.** Spécifie le langage de script activé pour le déclencheur. Toutes les actions **Execute script** que vous utilisez dans un même déclencheur utilisent le même langage.

Les options dans la section **Sauvegarder les Données Reçues** spécifient les commandes disponibles pour les données reçues par le déclencheur.

- **Sauvegarder les données reçues par le déclencheur dans un fichier.** Activez cette option pour sauvegarder les données reçues par le déclencheur. L'option **Variable** active le nom de fichier variable. Vous devez sélectionner une variable qui contient le chemin et le nom du fichier.
- **En cas d'erreur, sauvegarder les données reçues par le déclencheur dans un fichier.** Activez cette option pour sauvegarder les données dans le déclencheur uniquement si l'erreur se produit durant l'exécution de l'action. Vous allez activez cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème ultérieurement.

NOTE: NiceLabel Automation sauvegarde toujours les données reçues dans un nom de fichier temporaire, qui est effacé dès la fin d'exécution du déclencheur . La variable interne `DataFileName` pointe vers ce nom de fichier. Pour plus d'informations, consultez l'article Variables Internes.

Sécurité

- **Verrouiller et encoder le déclencheur.** Active la protection du déclencheur. Quand elle est activée, le déclencheur est verrouillé et ne peut pas être édité; les actions deviennent encodées.

Seul l'utilisateur ayant le mot de passe peut déverrouiller le déclencheur et le modifier.

Déclencheur De Serveur HTTP

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Pour en savoir plus sur les déclencheurs en général, consultez l'article Comprendre les Déclencheurs.

L'évènement d'activation HTTP survient quand les données sont reçues sur le socket surveillé (numéro d'adresse IP et de port). Au contraire de l'activateur TCP/IP, les données reçues ne sont pas un flux de données brutes mais doivent inclure l'entête standard HTTP. L'application tierce doit utiliser les méthodes de requête POST ou GET et procurer des données dans le corps du message ou dans la chaîne de requête. Le type de média Internet que vous utilisez dans le corps du message n'est pas important (Type MIME, ou Type Content). NiceLabel Automation recevra le message et vous pouvez définir un filtre pour extraire les données requises contenues dans le message.

Utilisation typique : Le système existant exécute une transaction, qui en fait envoie les données au serveur NiceLabel Automation formatées en message HTTP POST sur un socket spécifique. Le contenu de données peut être structuré en format CSV, XML et autres formats, ou il peut être structuré dans un format initial. Dans chaque cas, NiceLabel Automation va lire les données, analyser les valeurs en utilisant des filtres et imprimer les valeurs extraites sur les étiquettes. Pour plus de renseignements concernant l'analyse et l'extraction de données, consultez l'article Comprendre les Filtres.

Fourniture de données

Vous pouvez fournir les données pour l'activateur HTTP en utilisant une des méthodes suivantes. Vous pouvez également combiner les méthodes si nécessaire et utiliser les deux dans la même requête HTTP.

Données dans la chaîne de requête

Une chaîne de requête est la partie de l'URL (Localisateur de Ressource Uniforme) qui contient les données à passer au déclencheur HTTP.

Une URL typique contenant une chaîne (string) de requête (query) se présente comme suit :

```
http://serveur/chemin/?query_string
```

Le point d'interrogation est utilisé comme séparateur et ne fait pas partie de la chaîne de requête.

La chaîne de requête est habituellement composée d'une série de paires `nom:valeur`, dans laquelle chaque nom de champ et valeur sont séparés par un signe égal (=). Les séries de paires sont séparées par le signe et commercial (&). La chaîne de requête typique fournit les valeurs des champs (variables), comme suit :

```
champ1=valeur1&champ2=valeur2&champ3=valeur3
(field1=value1&field2=value2&field3=value3)
```

Le déclencheur HTTP contient le support d'extraction de valeurs de tous les champs et les sauvegarde dans les variables de même nom, donc vous ne devez pas définir des filtres pour extraire les valeurs de la chaîne de requête.

- Vous ne devez pas définir de variables dans le déclencheur pour les remplir avec les valeurs de la chaîne de requête. NiceLabel Automation extrait toutes les variables de la chaîne de requête et envoie leurs valeurs à l'étiquette actuellement active. Si des variables de même nom existent dans l'étiquette, leurs valeurs seront remplies. Si les variables n'existent pas dans l'étiquette, les valeurs seront ignorées et aucune erreur ne sera rapportée.
- Vous définirez les variables dans le déclencheur si vous avez besoin des valeurs pour certaines actions de ce déclencheur. Pour obtenir toutes les valeurs fournies dans la chaîne de requête, définissez simplement les variables qui ont le même nom de champs que les champs dans la chaîne de requête. Pour l'exemple ci-dessus, vous devrez définir les variables de déclencheurs avec les noms `champ1`, `champ2` et `champ3`.

Vous utiliserez généralement la méthode de requête GET HTTP pour procurer la chaîne de requête.

Données dans le corps de la requête HTTP

Vous devez utiliser la méthode de requête POST pour procurer le message dans le corps de la requête HTTP.

Vous pouvez envoyer n'importe quelles données et structure de données dans le corps, à condition que vous puissiez traiter les données dans les filtres NiceLabel Automation. Le contenu peut être formaté en XML, CSV, il peut être sous forme de texte, il peut même être sous forme de données binaires (encodage Base64). N'oubliez-pas que vous devrez analyser les données avec des filtres.

Si vous pouvez influencer la structure du message entrant, utilisez des structures standardisées, telles que XML ou CSV, pour simplifier la configuration du filtre.

Vous utiliserez la méthode de requête POST HTTP pour introduire les données dans le corps du message.

Général

Cette section vous permet de configurer les paramètres des déclencheurs de fichiers les plus importants.

- **Nom.** Spécifie le nom de déclencheur unique. Les noms vous aident à distinguer les différents déclencheurs quand vous les configurez dans Automation Builder et les exécutez ensuite dans Automation Manager.
- **Description.** Procure la possibilité de décrire la fonctionnalité de ce déclencheur. Vous pouvez l'utiliser pour écrire une courte description de la fonction du déclencheur.

Communication

NOTE: Ce déclencheur support le Protocole Internet version 6 (IPv6).

Cette section permet la configuration du numéro de port obligatoire et les options de feedback optionnelles. Vous pouvez utiliser les Codes de Réponse standard HTTP pour indiquer le succès de l'action d'exécution. Pour des actions plus avancées, vous pouvez également envoyer le contenu personnalisé à l'application procurant les données, soit dans une simple chaîne de retour ou en données binaires, tel qu'un aperçu de l'étiquette ou un flux d'impression.

L'URL typique pour se connecter au déclencheur HTTP est la suivante :

```
http://serveur:port/chemin/?query_string
```

- **Serveur**. Ceci est l'adresse IP ou FQDN de la machine où NiceLabel Automation est installé.
- **Port.** Spécifie le numéro de port sur lequel les données entrantes seront acceptées. Utilisez un numéro de port qui n'est pas utilisé par une autre application. Si le port sélectionné est utilisé, vous ne pourrez pas démarrer le déclencheur dans le Automation Manager. Pour plus d'informations concernant la sécurité, consultez l'article Sécuriser l'accès à vos déclencheurs.

  NOTE: Si votre serveur a l'hébergement-multiple activé (plusieurs adresses IP sur une ou plusieurs cartes réseau, NiceLabel Automation répondra au port défini pour toutes les adresses IP.

- **Chemin.** Spécifie le chemin optionnel dans l'URL. Cette fonctionnalité permet au NiceLabel Automation d'exposer multiples déclencheurs HTTP sur le même port. Le client utilise les déclencheurs au travers d'un même port dans une syntaxe de type REST, causant l'activation de différents déclencheurs par une URL différente. Si vous ne savez pas ce qu'il faut utiliser, laissez le chemin par défaut (\).
- **Connexion Sécurisée (HTTPS).** Active la couche de transport sécurisée pour vos messages HTTP et évite l'écoute clandestine. Pour plus d'informations d'implémentation, consultez l'article Utilisation de la Couche de Transport Sécurisée (HTTPS).

  Cette option est disponible dans NiceLabel Automation Enterprise.

- **Chaîne de requête.** Spécifie les paires nom-valeur dans l'URL. Un paramètre optionnel, les données sont habituellement fournies dans le corps de la requête HTTP.
- **Attendre la fin d'exécution du déclencheur.** Le protocole HTTP requiert du receveur (dans ce cas NiceLabel Automation) de renvoyer une réponse numérique à l'expéditeur indiquant l'état du message reçu. Par défaut, NiceLabel Automation répondra par le code 200, indiquant que les données ont bien été reçues, mais ceci ne donne pas d'informations concernant le succès des actions de déclenchement. Cette option spécifie que le déclencheur n'envoie pas la réponse immédiatement après la réception des données, mais attend que toutes les actions ont été exécutées et envoie ensuite le code de réponse indiquant le succès de l'action exécutée.

  Les codes de réponses HTTP sont les suivants :

Code de Réponse HTTP	Description
200	Toutes les actions ont été exécutées avec succès.
401	Nom d'utilisateur non-autorisé ou mot de passe erroné spécifié.
500	Des erreurs sont survenues durant l'action d'exécution.

  NOTE: Si vous désirez envoyer un feedback concernant le processus d'impression, assurez-vous d'activer le mode d'impression **synchrone**. Pour plus d'informations, Consultez l'article Mode d'impression Synchrone.

- **Nombre Maximum de requêtes concurrentes.** Spécifie le nombre maximum de connexions entrantes concurrentes. Autant de clients peuvent envoyer des données au déclencheur simultanément. Le nombre dépend aussi des performances physiques de votre serveur.
- **Type de Réponse.** Spécifie le type de votre message de réponse. Les types de médias Internet

fréquemment utilisés (aussi connus comme types MIME, ou types Content) sont prédéfinis dans le menu déroulant. Si votre type de média n'est pas disponible dans la liste, entrez le simplement vous-même. Les données de réponse seront envoyées à l'extérieur sous forme de feedback, formatées en type de média défini. L'option **Variable** active le type de média variable. Quand il est activé, vous devez sélectionner la variable qui contiendra le type de média.

> **NOTE:** Si vous ne spécifiez pas le Content-Type, NiceLabel Automation utilisera `application/octet-stream` par défaut.

- **Données de Réponse.** Définit le contenu du message de réponse. Vous pouvez utiliser le contenu fixe, un mixte de contenu fixe et variable, ou contenu variable seul. Pour insérer un contenu variable, cliquez le bouton avec une flèche à droite de la zone de données et insérez la variable de la liste contenant les données que vous voulez renvoyer comme message de réponse HTTP. Pour plus d'informations, consultez l'article Utiliser des valeurs composées.

  Certains exemples de ce que vous pouvez renvoyer comme réponse HTTP : messages d'erreur personnalisés, aperçu d'étiquette, fichiers PDF générés, fichier de flux d'impression (fichier spouleur), fichier XML avec les détails du générateur d'impression plus l'aperçu d'étiquette (encodé comme chaîne Base64), les possibilités sont infinies.

  > **NOTE:** Si vous allez seulement sortir un contenu binaire (tel qu'un aperçu de l'étiquette ou un flux d'impression), assurez-vous de sélectionner le type de média approprié, par ex. `image/jpeg` ou `application/octet-stream`.

Authentification d'Utilisateur

- **Activer l'authentification de base.** Spécifie que les messages entrant comprennent le nom d'utilisateur et mot de passe. Le déclencheur acceptera seulement les messages HTTP dont les informations d'identification correspondent à l'identification entrée ici. Pour plus d'informations concernant la sécurité, consultez l'article Sécuriser l'accès à vos déclencheurs.

Autre

Les options dans la section **Feedback du Moteur d'Impression** spécifient la communication avec le moteur d'impression.

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

- **Impression surveillée.** Active le mode d'impression synchrone. Utilisez cette option quand vous voulez renvoyer l'état de la tâche d'impression à l'application tierce. Pour plus d'informations, consultez l'article Mode d'impression Synchrone.

Les options dans la section **Traitement de Données** spécifient si vous voulez couper les données pour qu'elles entrent dans la variable ou ignorer les variables d'étiquettes manquantes. Par défaut, NiceLabel Automation rapporte les erreurs et interrompt le processus d'impression si vous essayez de sauvegarder une valeur trop longue dans la variable d'étiquette, ou essayez de paramétrer une valeur dans une variable d'étiquette inexistante.

- **Ignorer les contenus de variable excessifs.** Les valeurs de données dépassant la longueur de variable définie dans le concepteur d'étiquettes seront tronquées pour pouvoir entrer dans la

variable. Cette option s'applique lors du paramétrage de valeurs dans les filtres des fichiers de commande et au paramétrage de valeurs de variables de déclencheurs dans les variables d'étiquette de même nom.

> **EXEMPLE:** La variable d'étiquette accepte un maximum de 10 caractères. Quand cette option est activée, toute valeur de plus de 10 caractères sera tronquée au 10 premiers caractères, tous les caractères suivant le caractère numéro 10 seront ignorés.

- **Ignorer les variables d'étiquettes manquantes.** Quand vous effectuez l'impression avec des fichiers de commande (tel qu'un fichier JOB), le processus d'impression va ignorer toutes les variables qui sont spécifiées dans le fichier de commande (utilisant la commande SET), mais ne sont pas définies dans l'étiquette. Il n'y aura pas d'erreur lors de l'essai de paramétrage de la variable d'étiquette inexistante. Un processus semblable s'effectue quand vous définissez une zone d'affectation dans le filtre pour extraire toutes les paires *nom:valeur*, mais vous avez moins de variables définies dans l'étiquette.

Les options dans la section **Script** spécifient les possibilités de script.

- **Langage de Script.** Spécifie le langage de script activé pour le déclencheur. Toutes les actions **Execute script** que vous utilisez dans un même déclencheur utilisent le même langage.

Les options dans la section **Sauvegarder les Données Reçues** spécifient les commandes disponibles pour les données reçues par le déclencheur.

- **Sauvegarder les données reçues par le déclencheur dans un fichier.** Activez cette option pour sauvegarder les données reçues par le déclencheur. L'option **Variable** active le nom de fichier variable. Vous devez sélectionner une variable qui contient le chemin et le nom du fichier.
- **En cas d'erreur, sauvegarder les données reçues par le déclencheur dans un fichier.** Activez cette option pour sauvegarder les données dans le déclencheur uniquement si l'erreur se produit durant l'exécution de l'action. Vous allez activez cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème ultérieurement.

NOTE: NiceLabel Automation sauvegarde toujours les données reçues dans un nom de fichier temporaire, qui est effacé dès la fin d'exécution du déclencheur . La variable interne `DataFileName` pointe vers ce nom de fichier. Pour plus d'informations, consultez l'article Variables Internes.

Sécurité

- **Verrouiller et encoder le déclencheur.** Active la protection du déclencheur. Quand elle est activée, le déclencheur est verrouillé et ne peut pas être édité; les actions deviennent encodées. Seul l'utilisateur ayant le mot de passe peut déverrouiller le déclencheur et le modifier.

Déclencheur Web Service

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Enterprise**.

Pour en savoir plus sur les déclencheurs en général, consultez l'article Comprendre les Déclencheurs.

L'évènement d'activation Web Service survient quand les données sont reçues sur le socket surveillé (numéro d'adresse IP et de port). Les données doivent suivre les notations SOAP (données XML encodées dans les messages HTTP). L'interface du Web Service est décrit dans le document WSDL disponible avec chaque déclencheur Web Service défini. Web Service peut fournir un feedback à

propos de la tâche d'impression, mais vous devez activer le mode de traitement **synchrone**. Pour plus d'informations, consultez l'article Feedback de la tâche d'impression.

En général, Web Service est utilisé par des programmeurs pour intégrer l'impression d'étiquettes dans leurs applications. Le système existant exécute une transaction, qui en fait envoie les données au serveur NiceLabel Automation sur un socket spécifique, formatées comme un message SOAP. Les données peuvent être fournies en format CSV, XML et autres formats structurés, ou elles peuvent être fournies dans un format hérité. Dans chaque cas, NiceLabel Automation va lire les données, analyser les valeurs en utilisant des filtres et les imprimer sur les étiquettes. Pour plus de renseignements concernant l'analyse et l'extraction de données, consultez l'article Comprendre les Filtres.

Général

Cette section vous permet de configurer les paramètres des déclencheurs de fichiers les plus importants.

- **Nom.** Spécifie le nom de déclencheur unique. Les noms vous aident à distinguer les différents déclencheurs quand vous les configurez dans Automation Builder et les exécutez ensuite dans Automation Manager.
- **Description.** Procure la possibilité de décrire la fonctionnalité de ce déclencheur. Vous pouvez l'utiliser pour écrire une courte description de la fonction du déclencheur.

Communication

NOTE: Ce déclencheur support le Protocole Internet version 6 (IPv6).

Cette section permet la configuration du numéro de port obligatoire et les options de feedback optionnelles.

- **Port.** Spécifie le numéro de port sur lequel les données entrantes seront acceptées. Utilisez un numéro de port qui n'est pas utilisé par une autre application. Si le port sélectionné est utilisé, vous ne pourrez pas démarrer le déclencheur dans le Automation Manager. Pour plus d'informations concernant la sécurité, consultez l'article Sécuriser l'accès à vos déclencheurs.

  NOTE: Si votre serveur a l'hébergement-multiple activé (plusieurs adresses IP sur une ou plusieurs cartes réseau, NiceLabel Automation répondra au port défini pour toutes les adresses IP.

- **Connexion Sécurisée (HTTPS).** Active la couche de transport sécurisée pour vos messages HTTP et évite l'écoute clandestine. Pour plus d'informations d'implémentation, consultez l'article Utilisation de la Couche de Transport Sécurisée (HTTPS).
- **Nombre maximum d'appels concurrents.** Spécifie le nombre maximum de connexions acceptées. Autant de clients peuvent envoyer des données au déclencheur.
- **Données de Réponse.** Définit la réponse personnalisée qui peut être utilisée avec la méthode `ExecuteTriggerWithResponse` - exécuter le déclencheur avec réponse. Les données de réponse comprendront le contenu tel qu'il est fourni dans la zone de texte. Vous pouvez combiner des valeurs fixes, des valeurs variables et des caractères spéciaux. Pour insérer des

variables et des caractères spéciaux, cliquez le bouton avec une flèche à droite de la zone de texte. La réponse peut contenir des données binaires, telles qu'une image d'aperçu de l'étiquette et le fichier d'impression (*.PRN).

Feedback d'état

Par sa conception, le déclencheur Web Service procure le feedback concernant l'état des tâches d'impression créées. Le déclencheur acceptera les données fournies et les utilisera durant l'exécution des actions définies. L'exécution des actions peut être supervisée. Le déclencheur rapportera l'état de succès de chaque évènement d'exécution. Pour activer le rapportage d'état du processus d'impression, vous devez activer le Mode d'impression Synchrone.

Les méthodes (fonctions) suivantes sont exposées dans le déclencheur Web Service :

- **ExecuteTrigger. - exécuter le déclencheur** Cette méthode accepte l'introduction de données dans le traitement et procure le feedback de statut optionnel. Un des paramètres d'entrée active ou désactive le feedback. Si vous activez le rapportage d'état, le feedback contiendra l'ID d'erreur et une description détaillée de l'erreur. Si l'ID d'erreur est égale à 0, il n'y avait pas de problèmes durant la création du fichier d'impression. Si l'ID est supérieure à 0, une erreur est survenue durant le processus d'impression. Dans cette méthode, la réponse de Web Service n'est pas configurable et contiendra toujours l'ID d'erreur et la description de l'erreur.
- **ExecuteTriggerWithResponse. - exécuter le déclencheur avec réponse** Cette méthode accepte l'introduction de données dans le traitement et procure le feedback de statut personnalisé. La réponse de Web Service est configurable. Vous pouvez renvoyer toutes les données que vous voulez dans une des structures que vous avez. Vous pouvez utiliser des données binaires dans la réponse.
- **ExecuteTriggerAndSetVariables. - exécuter le déclencheur et définir les variables** Similaire à **ExecuteTrigger** (exécuter le déclencheur) ci-dessus, mais expose des paramètres entrants supplémentaires qui acceptent les listes de paires formatées *nom-valeur*. Le déclencheur va analyser automatiquement la liste, extraire les valeurs et les sauvegarder dans les variables de même nom; donc vous ne devez pas créer de filtre d'extraction vous-même.
- **ExecuteTriggerAndSetVariablesWithResponse. - exécuter le déclencheur et définir les variables avec réponse.** Similaire à **ExecuteTriggerWithResponse** (exécuter le déclencheur avec réponse) ci-dessus, mais expose des paramètres entrants supplémentaires qui acceptent les listes de paires formatées *nom-valeur*. Le déclencheur va analyser automatiquement la liste, extraire les valeurs et les sauvegarder dans les variables de même nom; donc vous ne devez pas créer de filtre d'extraction vous-même.

Pour plus d'informations concernant la structure des messages que vous pouvez envoyez à l'une ou l'autre méthode, consultez le chapitre WSDL ci-dessous.

WSDL

Spécifie le style des messages SOAP. Il peut être soit un **Remote Procedure Call (RPC)** ou de style **document**. Choisissez le style qui est supporté dans votre application procurant les données au NiceLabel Automation.

Le document WSDL (Web Service Description Language) définit les paramètres d'entrées et sorties de Web Service.

Si vous définissez un déclencheur Web Service sur le port 12345, déployez-le dans le Automation Manager et ensuite lancez-le, son WSDL sera disponible en :

```
http://localhost:12345
```

Le WSDL expose différentes méthodes qui fournissent toutes des données au déclencheur. Vous devrez choisir celui qui est le plus approprié pour ce que vous devez faire.

- Les méthodes ayant *WithResponse* (avec réponse) dans leurs noms vous permettent d'envoyer des réponses personnalisées, tels que des messages d'erreur personnalisés, des aperçus d'étiquette, des fichiers PDF, des fichiers d'impression (*.PRN) et similaires. Les méthodes sans *WithResponse* dans leur nom procureront le feedback, mais vous ne pouvez pas envoyer des réponses personnalisées. Le feedback contiendra des messages d'erreur par défaut.
- Les méthodes qui ont *SetVariables* (définir variables) dans leurs noms vous permettent de fournir une liste de variables en deux formats prédéfinis et leurs valeurs seront extraites et mappées automatiquement aux variables appropriées. Cela vous sauve du temps car vous ne devez pas définir un filtre pour effectuer l'extraction et le mappage. Pour les méthodes sans *SetVariables* dans leurs noms, vous devez définir le filtre vous-même.

L'interface Web Service définit les méthode suivantes :

Méthode ExecuteTrigger (exécuter le déclencheur)

La partie principale de la définition est la suivante :

```
<wsdl:message name="WebSrviTrg_ExecuteTrigger_InputMessage">
<wsdl:part name="text" type="xsd:string"/>
<wsdl:part name="wait" type="xsd:boolean"/>
</wsdl:message>

<wsdl:message name="WebSrviTrg_ExecuteTrigger_OutputMessage"
<wsdl:part name="ExecuteTriggerResult" type="xsd:int"/
<wsdl:part name="errorText" type="xsd:string"/>
</wsdl:message>
```

Il y a deux variables d'entrée (vous fournissez les valeurs) :

- **text.** (texte) Ceci est la chaîne de caractères d'entrée, qui peut être analysée par le filtre défini dans la configuration. Généralement la structure de la chaîne de caractères est CSV ou XML pour permettre une analyse facile par un filtre, mais vous pouvez utiliser n'importe quel format de texte.
- **wait.** (attendre) Ceci est un champ Booléen qui spécifie si vous allez attendre la réponse de l'état d'impression et si le Service Web doit fournir un feedback. Pour *Vrai* utilisez `1`, pour *Faux* utilisez `0`. Selon le type de méthode que vous sélectionnez, il y a soit une réponse prédéfinie ou vous pouvez envoyer la réponse personnalisée.

Il y a deux variables de sortie optionnelles (vous recevez leurs valeurs si vous les demandez) :

- **ExecuteTriggerResult - exécuter le résultat déclencheur.** La réponse intègre contiendra la valeur 0 s'il n'y a pas eu de problèmes de traitement des données, et elle contiendra une intègre plus grande que 0, quand une/des erreur(s) est/sont survenue(s). L'application qui exécute l'appel Web Service au NiceLabel Automation peut utiliser la réponse comme indicateur d'erreur.
- **errorText. - texte d'erreur** Cette valeur de chaîne contiendra la réponse d'état de la tâche d'impression. S'il n'y a pas d'erreurs durant le traitement de la tâche d'impression, la valeur sera une chaîne vide. Si une erreur est survenue, la valeur contiendra un message descriptif.

Méthode ExecuteTriggerWithResponse - exécuter le déclencheur avec réponse.

Cette méthode sera utilisée quand le déclencheur doit envoyer la réponse personnalisée après la fin de l'exécution.

Certains exemples de ce que vous pouvez renvoyer comme réponse : messages d'erreur personnalisés, aperçu d'étiquette, fichiers PDF générés, fichier de flux d'impression (fichier spouleur), fichier XML avec les détails du générateur d'impression plus l'aperçu d'étiquette (encodé comme chaîne Base64), les possibilités sont infinies.

La partie principale de la définition est la suivante :

```
<wsdl:message name="WebSrviTrg_ExecuteTriggerWithResponse_InputMessage">
<wsdl:part name="text" type="xsd:string"/>
<wsdl:part name="wait" type="xsd:boolean"/>
</wsdl:message>

<wsdl:message name="WebSrviTrg_ExecuteTriggerWithResponse_OutputMessage">
<wsdl:part name="ExecuteTriggerWithResponseResult" type="xsd:int"/>
<wsdl:part name="responseData" type="xsd:base64Binary"/>
<wsdl:part name="errorText" type="xsd:string"/>
</wsdl:message>
```

Il y a deux variables d'entrée (vous fournissez les valeurs) :

- **text.** (texte) Ceci est la chaîne de caractères d'entrée, qui peut être analysée par le filtre défini dans la configuration. Généralement la structure de la chaîne de caractères est CSV ou XML pour permettre une analyse facile par un filtre, mais vous pouvez utiliser n'importe quel format de texte.
- **wait.** (attendre) Ceci est un champ Booléen qui spécifie si vous allez attendre la réponse de l'état d'impression et si le Service Web doit fournir un feedback. Pour *Vrai* utilisez `1`, pour *Faux* utilisez `0`. Selon le type de méthode que vous sélectionnez, il y a soit une réponse prédéfinie ou vous pouvez envoyer la réponse personnalisée.

Il y a trois variables de sortie optionnelles :

- **ExecuteTriggerWithResponseResult - exécuter le déclencheur avec résultat réponse.** La réponse intègre contiendra la valeur 0 s'il n'y a pas eu de problèmes de traitement des données, et elle contiendra une intègre plus grande que 0, quand une/des erreur(s) est/sont survenue(s). L'application qui exécute l'appel Web Service au NiceLabel Automation peut utiliser la réponse comme indicateur d'erreur.
- **responseData. - donnée de réponse** La réponse personnalisée que vous pouvez définir dans la configuration du déclencheur Web Service.
- **errorText. - texte d'erreur** Cette valeur de chaîne contiendra la réponse d'état de la tâche d'impression. S'il n'y a pas d'erreurs durant le traitement de la tâche d'impression, la valeur sera une chaîne vide. Si une erreur est survenue, la valeur contiendra un message descriptif.

Méthode ExecuteTriggerAndSetVariables - exécuter le déclencheur et définir les varaiables

La partie principale de la définition est la suivante :

```
<wsdl:message name="WebSrviTrg_ExecuteTriggerAndSetVariables_InputMessage">
<wsdl:part name="text" type="xsd:string"/>
<wsdl:part name="variableData" type="xsd:string"/>
<wsdl:part name="wait" type="xsd:boolean"/>
</wsdl:message>
```

```
<wsdl:message name="WebSrviTrg_ExecuteTriggerAndSetVariables_OutputMessage">
<wsdl:part name="ExecuteTriggerAndSetVariablesResult" type="xsd:int"/>
<wsdl:part name="errorText" type="xsd:string"/>
</wsdl:message>
```

Il y a trois variables d'entrée (vous fournissez leurs valeurs) :

- **text.** (texte) Ceci est la chaîne de caractères d'entrée, qui peut être analysée par le filtre défini dans la configuration. Généralement la structure de la chaîne de caractères est CSV ou XML pour permettre une analyse facile par un filtre, mais vous pouvez utiliser n'importe quel format de texte.
- **wait.** (attendre) Ceci est un champ Booléen qui spécifie si vous allez attendre la réponse de l'état d'impression et si le Service Web doit fournir un feedback. Pour *Vrai* utilisez `1`, pour *Faux* utilisez `0`. Selon le type de méthode que vous sélectionnez, il y a soit une réponse prédéfinie ou vous pouvez envoyer la réponse personnalisée.
- **variableData.** Ceci est la chaîne de caractères contenant les paires *nom:valeur*. Le déclencheur va lire toutes les paires et assigner les valeurs fournies aux variables du déclencheur de même nom. Si la variable n'existe pas dans le déclencheur, la paire *nom:valeur* est éliminée. Quand vous fournissez la liste de variables et leurs valeurs par cette méthode, vous ne devez pas définir d'extraction de données avec les filtres. Le filtre se chargera de faire l'analyse.

  Le contenu de variableData peut être fourni dans une des deux structures disponibles.

  Structure XML

  Les variables sont fournies dans l'élément racine `<Variables />` du fichier XML. Le nom de variable est fourni avec le nom d'attribut, la valeur de variable est fournie par l'élément valeur.

  ```
  <?xml version="1.0" encoding="utf-8"?>
  <Variables>
  <variable name="Variable 1">Value 1</variable>
  <variable name="Variable 2">Value 2</variable>
  <variable name="Variable 3">Value 3</variable>
  </Variables>
  ```

  NOTE: Vous devrez incorporer vos données XML dans la section CDATA. **CDATA,** signifiant **données de caractère**, est une section de contenu d'élément qui est marquée pour que l'analyseur l'interprète seulement comme données de caractères, pas une balise. Tout le contenu est utilisé comme données de caractères, par exemple `<element>ABC</element>` sera interprété comme `<element>ABC</element>`. Une section CDATA commence par la séquence `<![CDATA[` et se termine par la séquence `]]>`. Placez simplement vos données XML entre ces séquences.

  Structure délimitée

  Les variables sont fournies dans un flux de données. Chaque paire *nom:valeur* est fournie sur une ligne séparée. Le nom de variable se situe à gauche du signe égal (=), la valeur de variable se situe à droite.

```
Variable 1=Valeur 1
Variable 2=Valeur 2
Variable 3=Valeur 3
```

Il y a deux variables de sortie optionnelles (vous recevez leurs valeurs si vous les demandez) :

- **ExecuteTriggerAndSetVariablesResult - exécuter le déclencheur et définir les varaiables.** La réponse intègre contiendra la valeur 0 s'il n'y a pas eu de problèmes de traitement des données, et elle contiendra une intègre plus grande que 0, quand une/des erreur(s) est/sont survenue(s). L'application qui exécute l'appel Web Service au NiceLabel Automation peut utiliser la réponse comme indicateur d'erreur.
- **errorText. - texte d'erreur** Cette valeur de chaîne contiendra la réponse d'état de la tâche d'impression. S'il n'y a pas d'erreurs durant le traitement de la tâche d'impression, la valeur sera une chaîne vide. Si une erreur est survenue, la valeur contiendra un message descriptif.

Méthode ExecuteTriggerAndSetVariablesWithResponse - exécuter le déclencheur avec réponse

Cette méthode sera utilisée quand le déclencheur doit envoyer la réponse personnalisée après la fin de l'exécution.

Certains exemples de ce que vous pouvez renvoyer comme réponse : messages d'erreur personnalisés, aperçu d'étiquette, fichiers PDF générés, fichier de flux d'impression (fichier spouleur), fichier XML avec les détails du générateur d'impression plus l'aperçu d'étiquette (encodé comme chaîne Base64), les possibilités sont infinies.

La partie principale de la définition est la suivante :

```
<wsdl:message name="WebSrviTrg_ExecuteTriggerAndSetVariablesWithResponse_
InputMessage">
<wsdl:part name="text" type="xsd:string"/>
<wsdl:part name="variableData" type="xsd:string"/>
<wsdl:part name="wait" type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="WebSrviTrg_ExecuteTriggerAndSetVariablesWithResponse_
OutputMessage">
<wsdl:part name="ExecuteTriggerAndSetVariablesWithResponseResult" type="xsd:int"/>
<wsdl:part name="responseData" type="xsd:base64Binary"/>
<wsdl:part name="errorText" type="xsd:string"/>
</wsdl:message>
```

Il y a trois variables d'entrée (vous fournissez leurs valeurs) :

- **text.** (texte) Ceci est la chaîne de caractères d'entrée, qui peut être analysée par le filtre défini dans la configuration. Généralement la structure de la chaîne de caractères est CSV ou XML pour permettre une analyse facile par un filtre, mais vous pouvez utiliser n'importe quel format de texte.
- **wait.** (attendre) Ceci est un champ Booléen qui spécifie si vous allez attendre la réponse de l'état d'impression et si le Service Web doit fournir un feedback. Pour *Vrai* utilisez `1`, pour *Faux* utilisez `0`. Selon le type de méthode que vous sélectionnez, il y a soit une réponse prédéfinie ou vous pouvez envoyer la réponse personnalisée.
- **variableData.** Ceci est la chaîne de caractères contenant les paires *nom:valeur*. Le déclencheur va lire toutes les paires et assigner les valeurs fournies aux variables du déclencheur de même nom. Si la variable n'existe pas dans le déclencheur, la paire *nom:valeur* est éliminée. Quand

vous fournissez la liste de variables et leurs valeurs par cette méthode, vous ne devez pas définir d'extraction de données avec les filtres. Le filtre se chargera de faire l'analyse.

Le contenu de variableData peut être fourni dans une des deux structures disponibles.

Structure XML

Les variables sont fournies dans l'élément racine `<Variables />` du fichier XML. Le nom de variable est fourni avec le nom d'attribut, la valeur de variable est fournie par l'élément valeur.

```
<?xml version="1.0" encoding="utf-8"?>
<Variables>
<variable name="Variable 1">Value 1</variable>
<variable name="Variable 2">Value 2</variable>
<variable name="Variable 3">Value 3</variable>
</Variables>
```

NOTE: Vous devrez incorporer vos données XML dans la section CDATA. **CDATA,** signifiant **données de caractère**, est une section de contenu d'élément qui est marquée pour que l'analyseur l'interprète seulement comme données de caractères, pas une balise. Tout le contenu est utilisé comme données de caractères, par exemple `<element>ABC</element>` sera interprété comme `<element>ABC</element>`. Une section CDATA commence par la séquence `<![CDATA[` et se termine par la séquence `]]>`. Placez simplement vos données XML entre ces séquences.

Structure délimitée

Les variables sont fournies dans un flux de données. Chaque paire *nom:valeur* est fournie sur une ligne séparée. Le nom de variable se situe à gauche du signe égal (=), la valeur de variable se situe à droite.

```
Variable 1=Valeur 1
Variable 2=Valeur 2
Variable 3=Valeur 3
```

Il y a trois variables de sortie optionnelles :

- **ExecuteTriggerAndSetVariablesWithResponseResult - exécuter le déclencheur avec résultat de réponse.** La réponse intègre contiendra la valeur 0 s'il n'y a pas eu de problèmes de traitement des données, et elle contiendra une intègre plus grande que 0, quand une/des erreur (s) est/sont survenue(s). L'application qui exécute l'appel Web Service au NiceLabel Automation peut utiliser la réponse comme indicateur d'erreur.
- **responseData. - donnée de réponse** La réponse personnalisée que vous pouvez définir dans la configuration du déclencheur Web Service.
- **errorText. - texte d'erreur** Cette valeur de chaîne contiendra la réponse d'état de la tâche d'impression. S'il n'y a pas d'erreurs durant le traitement de la tâche d'impression, la valeur sera une chaîne vide. Si une erreur est survenue, la valeur contiendra un message descriptif.

Autre

Les options dans la section **Feedback du Moteur d'Impression** spécifient la communication avec le moteur d'impression.

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

- **Impression surveillée.** Active le mode d'impression synchrone. Utilisez cette option quand vous voulez renvoyer l'état de la tâche d'impression à l'application tierce. Pour plus d'informations, consultez l'article Mode d'impression Synchrone.

Les options dans la section **Traitement de Données** spécifient si vous voulez couper les données pour qu'elles entrent dans la variable ou ignorer les variables d'étiquettes manquantes. Par défaut, NiceLabel Automation rapporte les erreurs et interrompt le processus d'impression si vous essayez de sauvegarder une valeur trop longue dans la variable d'étiquette, ou essayez de paramétrer une valeur dans une variable d'étiquette inexistante.

- **Ignorer les contenus de variable excessifs.** Les valeurs de données dépassant la longueur de variable définie dans le concepteur d'étiquettes seront tronquées pour pouvoir entrer dans la variable. Cette option s'applique lors du paramétrage de valeurs dans les filtres des fichiers de commande et au paramétrage de valeurs de variables de déclencheurs dans les variables d'étiquette de même nom.

  EXEMPLE: La variable d'étiquette accepte un maximum de 10 caractères. Quand cette option est activée, toute valeur de plus de 10 caractères sera tronquée au 10 premiers caractères, tous les caractères suivant le caractère numéro 10 seront ignorés.

- **Ignorer les variables d'étiquettes manquantes.** Quand vous effectuez l'impression avec des fichiers de commande (tel qu'un fichier JOB), le processus d'impression va ignorer toutes les variables qui sont spécifiées dans le fichier de commande (utilisant la commande SET), mais ne sont pas définies dans l'étiquette. Il n'y aura pas d'erreur lors de l'essai de paramétrage de la variable d'étiquette inexistante. Un processus semblable s'effectue quand vous définissez une zone d'affectation dans le filtre pour extraire toutes les paires *nom:valeur*, mais vous avez moins de variables définies dans l'étiquette.

Les options dans la section **Script** spécifient les possibilités de script.

- **Langage de Script.** Spécifie le langage de script activé pour le déclencheur. Toutes les actions **Execute script** que vous utilisez dans un même déclencheur utilisent le même langage.

Les options dans la section **Sauvegarder les Données Reçues** spécifient les commandes disponibles pour les données reçues par le déclencheur.

- **Sauvegarder les données reçues par le déclencheur dans un fichier.** Activez cette option pour sauvegarder les données reçues par le déclencheur. L'option **Variable** active le nom de fichier variable. Vous devez sélectionner une variable qui contient le chemin et le nom du fichier.
- **En cas d'erreur, sauvegarder les données reçues par le déclencheur dans un fichier.** Activez cette option pour sauvegarder les données dans le déclencheur uniquement si l'erreur se produit durant l'exécution de l'action. Vous allez activez cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème ultérieurement.

NOTE: NiceLabel Automation sauvegarde toujours les données reçues dans un nom de fichier temporaire, qui est effacé dès la fin d'exécution du déclencheur . La variable interne `DataFileName` pointe vers ce nom de fichier. Pour plus d'informations, consultez l'article Variables Internes.

Sécurité

- **Verrouiller et encoder le déclencheur.** Active la protection du déclencheur. Quand elle est activée, le déclencheur est verrouillé et ne peut pas être édité; les actions deviennent encodées. Seul l'utilisateur ayant le mot de passe peut déverrouiller le déclencheur et le modifier.

Utilisation De Variables

Variables

Les variables sont utilisées comme conteneurs de valeurs de données. Vous avez besoin de variables pour transférer les valeurs à l'étiquette dans l'action **Print Label**, ou pour utiliser les valeurs dans d'autres actions de manipulations de données. Typiquement, le filtre va extraire les valeurs des flux de données reçues par le déclencheur et envoyer les valeurs dans les variables. Pour plus de renseignements, consultez l'article Comprendre les Filtres.

Généralement, vous voulez envoyer les valeurs des variables au modèle d'étiquette et imprimer l'étiquette. Le mécanisme pour envoyer des valeurs de variables aux étiquettes utilise le mappage de nom automatique - la valeur de variable définie dans le déclencheur sera envoyée à la variable définie dans l'étiquette de même nom. Vous pouvez définir les variables de trois façons :

- **Importer les variables du fichier d'étiquette.** Pour le mappage automatique expliqué ci-dessus, il est conseillé d'importer chaque fois les variables de l'étiquette. Cette action assure que les noms de variables correspondent et gagne du temps. La variable importée n'hérite pas seulement du nom de variable, mais aussi des caractéristiques supportées de la variable, telles que la longueur et valeur par défaut.
- **Définition manuelle des variables.** Quand vous définissez les variables manuellement, vous devez être particulièrement attentif à l'utilisation de noms identiques aux variables dans l'étiquette. Vous devez définir manuellement les variables qui n'existent pas dans l'étiquette, mais vous en avez besoin dans le déclencheur.

  NOTE: Un exemple de variables seraient `LabelName`, `PrinterName`, `Quantity` et autres variables similaires dont vous devez mémoriser le nom d'étiquette, le nom d'imprimante, la quantité et autres méta-valeurs assignées par le filtre.

- **Activer les variables internes.** Les valeurs pour les valeurs internes sont assignées par le NiceLabel Automation et sont disponibles comme valeurs en lecture seule. Pour plus d'informations, consultez l'article Variables Internes.

Si vous activez la **zone d'affection** (dans les filtres XML et Texte Non-structuré) et **structure dynamique** (dans filtre de Texte Structuré), NiceLabel Automation le filtre va extraire les paires **nom:valeur** des données du déclencheur et envoyer automatiquement les valeurs aux variables de même nom qui sont définies dans l'étiquette. Le mappage manuel des variables n'est pas nécessaire.

Caractéristiques

- **Nom.** Spécifie le nom de variable unique. Les noms ne sont pas sensibles à la casse. Même si vous pouvez utiliser des espaces dans les noms, il est conseillé de les éviter. C'est encore plus important quand vous utilisez les variables dans des scripts ou conditions d'actions, car vous devrez les entourer de crochets.

- **Caractères autorisés.** Spécifie la liste de caractères que la valeur peut avoir. Vous avez le choix entre *Tout* (tous les caractères sont acceptés), Numérique (seuls les chiffres sont acceptés), et Binaire (tous les caractères et codes de contrôle sont acceptés).
- **Longueur limite de variable.** Spécifie le nombre maximal de caractères que la la variable peut occuper.
- **Longueur fixe.** Spécifie que la valeur doit occuper exactement le nombre de caractères défini par sa longueur.

  NOTE: Vous devez limiter la longueur de variable pour certains objets sur l'étiquette, tel que le code à barres EAN-13, qui accepte 13 caractères.
- **Valeur requise.** Spécifies que la variable doit contenir une valeur.
- **Valeur par défaut.** Spécifie une valeur par défaut. La valeur par défaut sera toujours utilisée si aucune valeur n'est assignée à la variable.

Utiliser Des Valeurs Composées

Certains objets dans la configuration du déclencheur acceptent les valeurs composées. Le contenu peut être un mélange de valeurs fixes, variables de caractères spéciaux (codes de contrôle). Les objets qui acceptent les valeurs composées sont identifiés par un petit bouton à flèche sur le côté droit de l'objet. Vous pouvez cliquer sur le bouton à flèche pour insérer soit une variable ou un caractère spécial.

- **Utilisation de valeurs fixes.** Vous pouvez introduire la valeur fixe pour la variable.

```
Ceci est une valeur fixe.
```

- **Utiliser des valeurs fixes et des données de variables.** Vous pouvez aussi définir les valeurs combinées, composées de valeurs variables et fixes. Les noms de variable doivent être entourés par des crochets `[]`. Vous pouvez introduire les variables manuellement, ou les insérer en cliquant sur le bouton à flèche sur la droite. Durant le traitement, les variables seront fusionnées ensembles avec les données fixes et utilisée comme contenu. Pour plus d'informations, consultez l'article Conseils et astuces d'utilisation de Variables dans les Actions.
  Dans ce cas, le contenu sera fusionné de trois variables et de certaines données fixes.

```
[variable1] // Ceci est une valeur fixe [variable2][variable3]
```

- **Utiliser les caractères spéciaux.** Vous pouvez également ajouter des caractères spéciaux au mélange. Vous pouvez introduire les caractères spéciaux manuellement, ou les insérer. Pour plus d'informations, consultez l'article Introduction de Caractères Spéciaux (Codes de Contrôle).
  Dans ce cas, la valeur de la `variable1` sera fusionnée avec des données fixes et de caractères binaires de saut de page.

```
[variable1] Saut de page va suivre ce texte fixe <FF>
```

Variables Internes

Les variables internes sont pré-définies par NiceLabel Automation. Leurs valeurs sont assignées automatiquement et sont seulement disponibles en mode de lecture. L'icône avec un symbole de

cadenas devant le nom de variable distingue les variables internes des variables définies par l'utilisateur. Vous pouvez utiliser les variables internes dans vos actions de la même façon que vous utilisez les variables définies par l'utilisateur. Les variables internes des déclencheurs sont internes pour chaque déclencheur.

Variable Interne	Disponible dans le déclencheur	Description
ActionLastErrorDesc	Tout	Fournit la description de la dernière erreur survenue. Vous pouvez utiliser cette valeur dans un retour au système hôte pour identifier la cause du défaut.
ActionLastErrorID	Tout	Fournit l'ID de la dernière erreur survenue. C'est une valeur intègre. Quand la valeur est 0, il n'a pas eu d'erreur. Vous pouvez utiliser cette valeur dans les conditions, pour évaluer si une erreur est survenue ou pas.
BytesOfReceivedData	TCP/IP	Fournit le nombre d'octets reçus par le déclencheur.
ComputerName	Tout	Fournit le nom de l'ordinateur où la configuration s'exécute.
ConfigurationFileName	Tout	Fournit le chemin et nom de fichier de la configuration actuelle (fichier .MISX).
ConfigurationFilePath	Tout	Fournit le chemin du fichier de configuration actuel. Voir aussi la description de ConfigurationFileName.
DataFileName	Tout	Fournit le chemin et nom de fichier de la copie de travail des données reçues. Chaque fois que le déclencheur accepte les données, il en fait une copie de sauvegarde vers le nom de fichier unique identifié par cette variable.
Base de données	Base de données	Fournit le type de base de données comme configuré dans le déclencheur.
Date	Tout	Fournit la date actuelle dans le format spécifié par le système locale, telle que 26.2.2013.
DateDay	Tout	Fournit le nombre actuel du jour du mois, tel que 26.
DateMonth	Tout	Fournit le nombre actuel du mois de l'année, tel que 2.
DateYear	Tout	Fournit le nombre actuel de l'année, tel que 2013.
DefaultPrinterName	Tout	Fournit le nom du pilote d'imprimante qui est défini par défaut.
DriverType	Base de données	Fournit le nom du pilote utilisé pour la connexion à la base de données spécifiée.
Hostname	TCP/IP	Fournit le nom de l'appareil/de l'ordinateur qui se connecte au déclencheur.
HttpMethodName	HTTP	Fournit le nom de méthode que l'utilisateur a fourni dans la requête HTTP.
HttpPath	HTTP	Fournit le chemin défini dans le déclencheur HTTP.
HttpQuery	HTTP	Fournit le contenu de la chaine de requête telle que reçue par le déclencheur HTTP.
NumberOfRowsReturned	Base de données	Fournit le nombre de rangées que le déclencheur reçoit d'une base de données.
LocalIP	TCP/IP	Fournit l'adresse IP locale sur laquelle le déclencheur a répondu. C'est utile si vous avez une machine d'hébergement multiple avec plusieurs cartes d'interface réseau (NIC) et voulez déterminer à quelle adresse IP le client s'est connecté. C'est utile en cas de scénario de remplacement de l'imprimante.

PathDataFileName	Tout	Fournit le chemin dans la variable DataFileName, sans le nom du fichier. Voir aussi la description de DataFileName.
PathTriggerFileName	Fichier	Fournit le chemin dans la variable TriggerFileName, sans le nom du fichier. Voir aussi la description de TriggerFileName.
Port.	TCP/IP, HTTP, Service Web	Fournit le numéro de port comme défini dans le déclencheur.
RemoteHttpIp	HTTP	Fournit le nom de l'appareil/de l'ordinateur qui se connecte au déclencheur.
RemoteIP	Service Web	Fournit le nom de l'appareil/de l'ordinateur qui se connecte au déclencheur.
ShortConfigurationFileName	Tout	Fournit le nom de fichier du fichier de configuration file, sans le chemin. Voir aussi la description de ConfigurationFileName.
ShortDataFileName	Tout	Fournit le chemin vers la variable DataFileName, sans le chemin. Voir aussi la description de DataFileName.
ShortTriggerFileName	Fichier	Fournit le nom du fichier vers la variable TriggerFileName, sans le chemin. Voir aussi la description de TriggerFileName.
SystemUserName	Tout	Fournit le nom Windows de l'utilisateur connecté.
TableName	Base de données	Fournit les nom de la table comme utilisée dans le déclencheur.
Heure	Tout	Fournit l'heure actuelle dans le format spécifié par le système local, telle que 15:18.
TimeHour	Tout	Fournit la valeur de l'heure actuelle, tel que 15.
TimeMinute	Tout	Fournit la valeur de minutes actuelle, tel que 18.
TimeSecond	Tout	Fournit la valeur de secondes actuelle, tel que 25.
TriggerFileName	Fichier	Fournit le nom du fichier qui a déclenché les actions. C'est utile quand vous surveillez un ensemble de fichiers dans le dossier, pour identifier exactement le fichier qui a déclenché les actions.
TriggerName	Tout	Fournit le nom du déclencheur comme défini par l'utilisateur.
Nom d'utilisateur.		Fournit le nom d'utilisateur NiceLabel Automation de l'utilisateur actuellement connecté. La variable a un contenu si la connexion d'utilisateur est activée.

Variables Globales

Les variables globales sont un type de variables qui peuvent être utilisées sur des étiquettes différentes. Les variables globales sont définies en dehors du fichier d'étiquette et mémorisent la dernière valeur utilisée. Les variables globales sont typiquement définies comme des compteurs globaux. La variable globale va fournir une valeur unique pour chaque étiquette demandant une nouvelle valeur. Un verrouillage du fichier est utiliser, pour assurer l'unicité de chaque valeur.

Les variables globales sont définies dans le concepteur d'étiquettes, NiceLabel Automation va seulement les utiliser. Si votre NiceLabel Automation est installé sur un ordinateur différent que le concepteur NiceLabel, vous devrez copier le fichier de définition des variables globales sur la machine NiceLabel Automation où la production d'impression d'étiquettes se déroule.

Pour copier les variables globales vers la machine NiceLabel Automation, effectuez les opérations suivantes :

1. Sur l'ordinateur avec le concepteur NiceLabel, naviguez vers le dossier suivant :

   Sur Windows Vista, Windows 7, Windows Server 2008, Windows Server 2012 et supérieur

   ```
   %PROGRAMDATA%\EuroPlus\Variables
   ```

   Sur Windows XP, Windows Server 2003

   ```
   %ALLUSERSPROFILE%\EuroPlus\Variables
   ```

2. Copiez les fichiers `GLOBAL.TDB` et `GLOBALS.TDB.SCH`.
3. Collez les fichiers dans le même dossier sur l'ordinateur NiceLabel Automation.

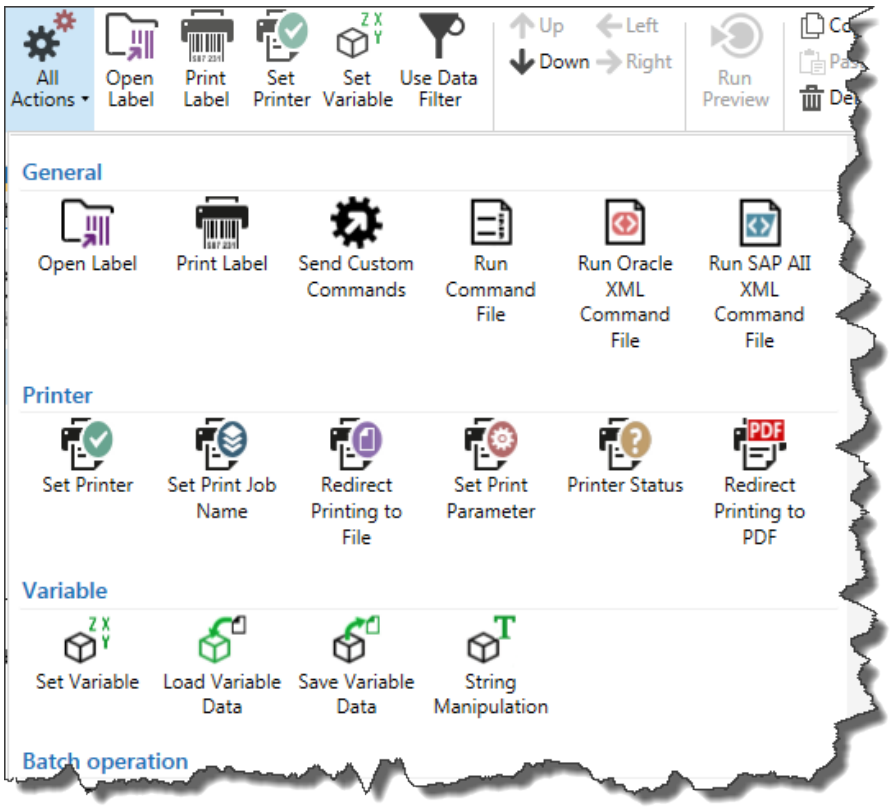
Utilisation Des Actions

Actions

La section Actions spécifie la liste d'actions qui seront exécutées chaque fois que le déclencheur est activé.

Définir les actions.

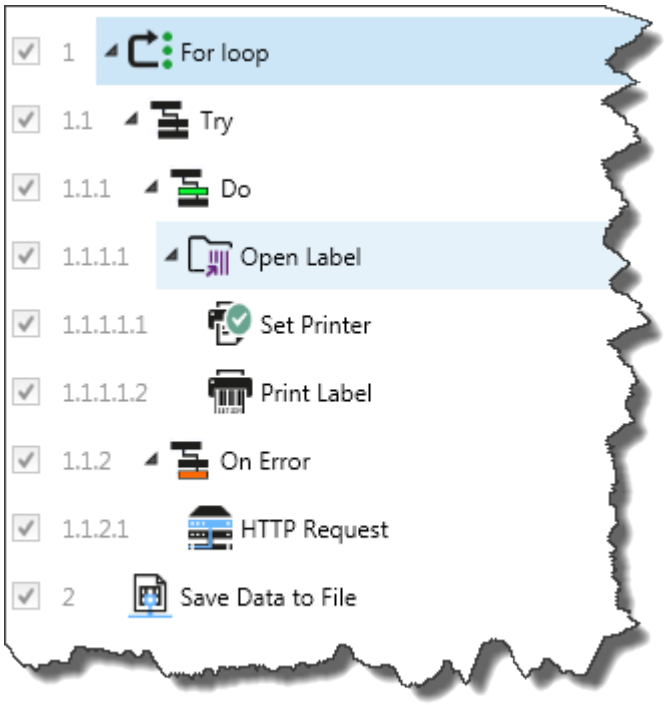
Pour définir l'action, cliquez sur l'icône action dans le groupe du ruban Insérer une Action. Le ruban principal contient les actions les plus souvent utilisées. Pour voir toutes les actions disponibles, cliquez sur le bouton **Toutes les Actions**. Pour voir les commandes disponibles pour l'action sélectionnée, effectuez un clic-droit pour sélectionner une commande de la liste.



Actions imbriquées.

Certaines actions ne peuvent pas être utilisées seules. Leur fonctionnalité spécifique requiert qu'elles

soient imbriquées sous une autre action. Utilisez les boutons du groupe de ruban **Ordre d'Action** pour changer le placement de l'action. Chaque action est identifiée par le numéro qui montre sa position dans la liste, y compris l'imbrication. Ce numéro ID sera affiché dans le message d'erreur, ce qui vous permettra de retrouver plus facilement l'action problématique.



NOTE: L'action **Imprimer l'Étiquette** est un bon exemple d'une telle action. Vous devez la positionner sous l'action **Ouvrir l'Étiquette**, pour qu'elle se réfère à l'étiquette exacte à imprimer.

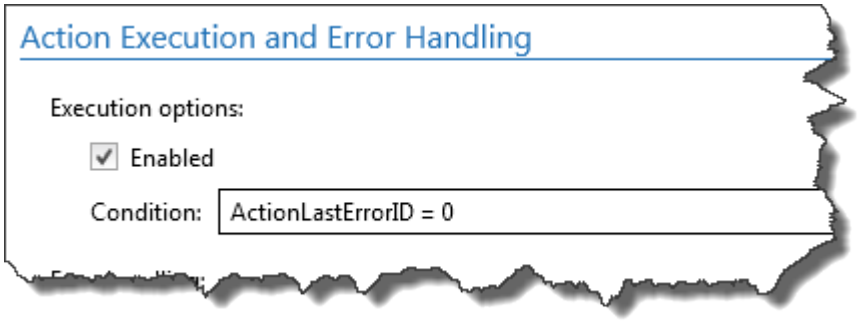
Exécution d'Action.

Les actions dans la liste ne s'exécuteront qu'une fois par déclencheur. L'exécution de l'action se fait de haut en bas, donc l'ordre des actions est important. Il y a deux exceptions. Les actions **For Loop (en boucle)** et **Use Data Filter** (utiliser un filtre de données) exécuteront les actions imbriquées plusieurs fois. For loop (en boucle) aussi souvent que défini dans ses caractéristiques, Use Data Filter (utiliser un filtre de données) aussi souvent qu'il y a d'enregistrements dans le bloc de données retourné par le filtre associé.

NiceLabel Automations'exécute comme un service sous un compte utilisateur Windows spécifique et hérite les permissions de sécurité du compte. Pour plus d'informations, consultez l'article Fonctionnement en Mode de Service.

Actions conditionnelles.

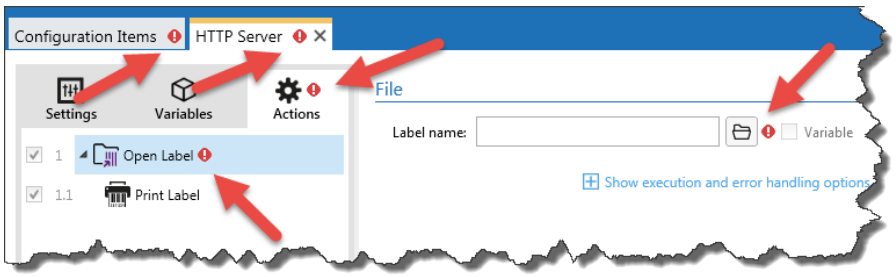
Chaque action peut être conditionnelle. Un action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. La condition est une ligne de script (VBScript ou Python). Pour définir la condition, cliquer **Afficher les options d'exécution et de traitement d'erreurs** dans les caractéristiques d'action pour développer les possibilités.



Dans ce cas, l'action sera seulement exécutée si l'action précédente s'est terminée avec succès, donc la variable interne `ActionLastErrorID` a la valeur 0. Pour utiliser une telle condition avec des variables internes, vous devez d'abord activer la variable interne.

Identifier les actions qui sont dans l'erreur de configuration.

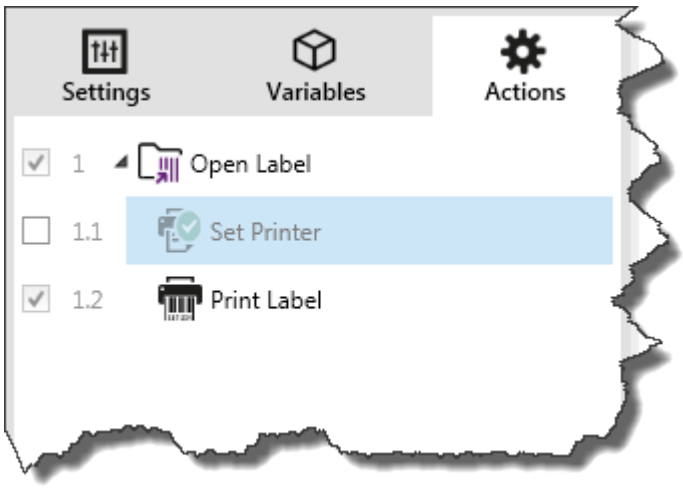
Si l'action n'est pas configurée complètement, elle sera marqué d'un icône avec un point d'exclamation rouge. Une telle action Ne peut pas être exécutée. Vous pouvez inclure une telle action dans la liste, mais vous devrez terminer la configuration avant de pouvoir lancer le déclencheur. Si une des actions imbriquées est en erreur, toutes les flèches de développement parent (à gauche du nom de l'action) seront également colorées en rouge, pour indiquer l'erreur de sous-action.



Dans ce cas, l'action Ouvrir l'Étiquette est dans l'erreur de configuration. Il n'y a pas de paramètre spécifié pour le nom d'étiquette. le point d'exclamation rouge s'affiche à côté du paramètre erroné dans l'action, dans la liste d'actions, dans l'onglet Actions, dans l'onglet Déclencheur et dans l'onglet Éléments de Configuration, pour faciliter son identification.

Désactiver les actions.

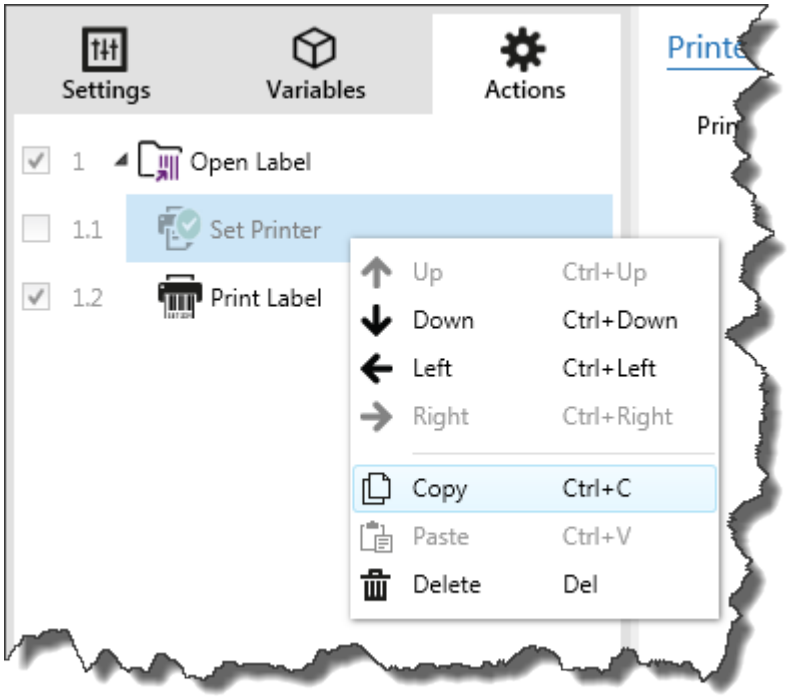
Par défaut, chaque action nouvellement créée est active et sera exécutée au lancement du déclencheur. Vous pouvez désactiver les actions dont vous n'avez pas besoin, mais que vous voulez garder dans la configuration. La case à cocher devant le nom de l'action dans la liste d'actions définies est un raccourci pour activer/désactiver l'action.



Dans ce cas, l'action Définir l'Imprimante est encore définie dans la liste d'actions mais a été désactivée. Elle n'est actuellment pas nécessaire et sera ignorée durant l'exécution, mais vous pouvez facilement la réactiver.

Copier l'action.

Vous pouvez copier l'action et la coller sur le même ou sur un autre déclencheur. Vous pouvez utiliser les raccourcis-clavier Windows standards ou effectuer un clic-droit sur l'action.



Un clic-droit sur l'action affichera les raccourcis de commandes disponibles pour l'objet actuellement sélectionné.

Naviguer dans la liste d'actions.

Vous pouvez utiliser la souris pour sélectionner l'action définie et puis cliquer sur le bouton à flèche correspondant dans le groupe **Ordre d'Action** du ruban. Vous pouvez également utiliser le clavier. Les touches du curseur vont déplacer la sélection dans la liste d'actions, les touches Ctrl + clés curseur vont déplacer la position de l'action vers le haut ou le bas, et à gauche ou à droite pour l'imbrication.

Effacer Un Fichier

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Efface le fichier sur le disque. NiceLabel Automation fonctionne comme un service sous un compte Windows défini. Assurez-vous que le compte a les permissions pour effacer le fichier dans le dossier spécifié.

Fichier

- **Nom de fichier.** Spécifie le chemin et nom du fichier. Il peut être codé-en-dur, et le même fichier sera utilisé à chaque fois. Si vous n'utilisez que le nom de fichier sans le chemin, le dossier où est sauvegardé le fichier de configuration (.MISX) sera utilisé. Vous pouvez utiliser une référence relative au nom de fichier, où le dossier avec le fichier .MISX est utilisé comme dossier racine. L'option **Variable** active le nom de fichier variable. Vous pouvez sélectionner une variable simple qui contient le chemin et/ou le nom de fichier ou vous pouvez combiner différentes variables qui vont créer le nom de fichier. Pour plus d'informations, consultez l'article Utiliser des valeurs composées.

  NOTE: Utilisez la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  EXEMPLE: A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

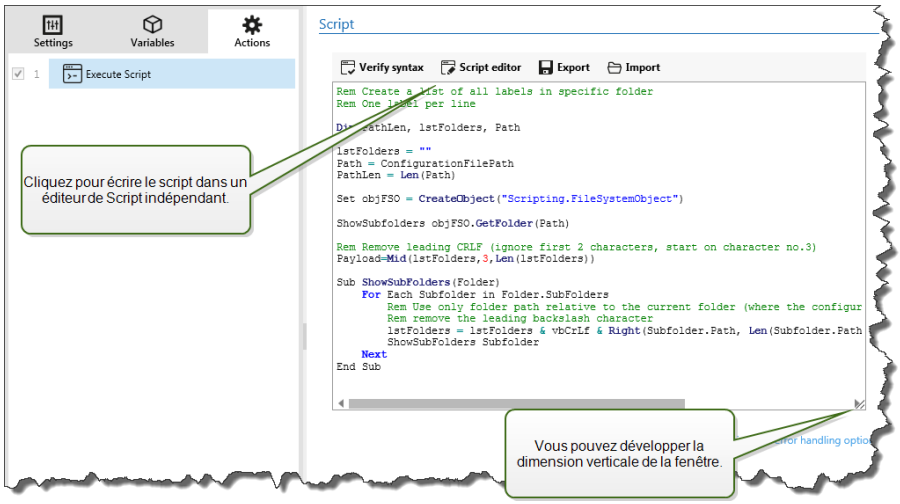
- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Exécuter Un Script

Améliore les fonctionnalités du logiciel en utilisant les scripts VBScript ou Python personnalisés. Vous pouvez utiliser cette fonction si les actions incorporées ne correspondent pas aux besoins de

manipulations de données. NiceLabel Automation fonctionne comme une application de service et comme telle, n'a pas accès au bureau; vous ne pouvez donc pas utiliser des fonctions telles que `MsgBox()` pour interagir avec le bureau dans VBScript. Assurez-vous aussi que le compte Windows sous lequel vous utilisez le service a les privilèges pour exécuter les commandes dans le script. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

Le support de VBScript est déjà disponnible avec votre système Windows. Pour installer le support Python, consultez l'article KB249 de la Base de Connaissances.



NOTE: Le type de script est configuré par déclencheur dans les caractéristiques du déclencheur. Toutes les actions du Exécuter le Script dans un déclencheur doivent être de même type.

Script

Définit le script à exécuter. Vous pouvez l'éditeur à-l'écran, ou lancer l'**Éditeur de Script** externe. L'éditeur externe contient également la référence pour toutes les fonctions et objets de script disponibles. NiceLabel Automation inclut certaines fonctions complémentaires, telles que des algorithmes de contrôle numérique pour différents codes à barres. Les fonctions complémentaires sont accessibles depuis l'Éditeur de Script.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  EXEMPLE: A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action

d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Exécuter Une Instruction SQL

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Envoie des commandes SQL vers un serveur SQL et collecte les résultats. Utilisez les commandes SELECT, INSERT, UPDATE, et DELETE.

Cette action s'utilise à deux feins.

- **Obtention de données additionnelles de la base de données.** Le déclencheur recevra les données pour l'impression de l'étiquette, mais pas toutes les données requises. Par exemple, le déclencheur reçoit la valeur pour `ID Produit` et `Description`, mais pas le `Prix`. Nous devons rechercher la valeur de `Price` (prix) dans la base de données SQL.

  Voir l'exemple de code SQL

  ```
  SELECT Price FROM Products
   (sélectionner Prix de Produits) Où ID = :[Product ID]
  ```

  L'`ID` est le champ dans la base de données, `Product ID` est une variable définie dans le déclencheur.

- **Mettre à jour ou effacer des données de la base de données.** Quand l'étiquette est imprimée, vous allez mettre à jour l'enregistrement dans la base de données en signalant au système que les enregistrements particuliers ont déjà été traités.

  Voir l'exemple de code SQL

  Vous changerez le champ de la table `AlreadyPrinted` (déjà imprimé) à `True` (vrai) pour l'enregistrement actuellement traité.

  ```
  UPDATE Products
  SET AlreadyPrinted = True
  WHERE ID = :[Product ID] (où ID =)
  ```

  Ou vous allez effacer l'enregistrement actuel de la base de données, car il n'est plus nécessaire.

  ```
  DELETE FROM Products
  WHERE ID = :[Product ID]
  ```

  L'`ID` est le champ dans la base de données, `Product ID` est une variable définie dans le déclencheur.

NOTE: Pour utiliser la valeur d'une variable dans une instruction SQL, vous devez utiliser le signe double-point (:) devant le nom. Ceci signale au NiceLabel Automation que les noms de variable suivent.

Connexion base de données.

Définit les paramètres de connexion à la base de données. Avant de pouvoir envoyer une phrase SQL à la base de données, vous devez paramétrer sa connexion. Cliquez le bouton Définir et suivez les instructions à l'écran. Vous pouvez vous connecter à une source de données qui peut être contrôlée par des instructions SQL, donc vous ne pouvez pas utiliser de fichiers texte (CSV) et Excel.

Instructions SQL

Définit l'instruction SQL --or query--to sera exécuté. Vous pouvez utiliser des instructions de Data Manipulation Language (DML) pour exécuter les requêtes aux tables de base de données existantes. Vous pouvez utiliser les instructions SQL standards, telles que SELECT, INSERT, DELETE et UPDATE, y compris JOINS, fonction et mots clés. Vous ne pouvez pas utiliser les instructions de Data Definition Language (DDL) pour créer des bases de données et des tables (CREATE DATABASE, CREATE TABLE), ou pour les effacer (DROP TABLE).

Cliquez le bouton **Test** dans la barre d'outils pour ouvrir la section Aperçu de Données, dans laquelle vous pouvez tester l'exécution de l'instruction SQL. Cliquez le bouton **Insérer une Variable** pour insérer les variables du déclencheur dans l'instruction SQL. Le contrôle d'éditeur vous permet d'**Exporter/Importer** l'instruction SQL vers/d'un fichier.

Mode Exécution.

Spécifie le mode d'exécution explicite. Pour certaines requêtes SQL complexes, il devient de plus en plus difficile de déterminer automatiquement ce qui est l'action supposée. Si la logique intégrée a des problèmes pour identifier vos intentions, sélectionnez l'action principale manuellement.

- **Automatique.** L'application essaiera de déterminer l'action automatiquement.
- **Renvoie un ensemble de données (SELECT).** Vous prévoyez la réception d'un ensemble de données avec enregistrement.
- **Ne renvoie pas un ensemble de données avec enregistrement (INSERT - insérer, DELETE - effacer, UPDATE - mettre à jour).** Vous exécutez une requête qui ne renverra pas de données. Vous voulez soit insérer un nouvel enregistrement, effacer ou mettre à jour des enregistrements existants. Le résultat sera une réponse d'état concernant le nombre de lignes affectées par votre requête.

Sauver le Résultat dans une Variable

Définit la variable qui va sauvegarder le résultat de l'instruction SQL.

- **Résultat de l'instruction SELECT.** Quand vous exécutez l'instruction SELECT, le résultat sera un ensemble d'enregistrements. Vous recevrez le contenu de texte en format CSV. La première ligne contiendra les noms de champs renvoyés comme résultats. Les ligne suivantes contiendront les enregistrements.

Pour extraire les valeurs des ensembles de données renvoyées et les utiliser dans d'autres actions, définissez le Configuration de Filtre de Texte Structuré et exécutez l'action Utilisation d'un Filtre de Données sur le contenu de cette variable.

- **Résultat des instructions INSERT, DELETE et UPDATE.** Quand vous utilisez les instructions INSERT, DELETE et UPDATE, le résultat sera un chiffre indiquant le nombre d'enregistrements affectés dans la table.

Aperçu des données

Cette section vous permet de tester l'exécution de vos instructions SQL avec des données réelles. Pour protéger les données d'une mise à jour accidentelle, assurez-vous que l'option **Simuler l'exécution** est activée. Les instructions INSERT - insérer, DELETE - effacer, UPDATE - mettre à jour seront exécutées, ce qui vous donnera un retour sur le nombre de données affectées, ensuite la transactions sera annulée.

Si vous utilisez des variables de déclenchement dan l'instruction SQL, vous serez en mesure d'entrer leurs valeurs pour l'exécution du test.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  EXEMPLE: A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

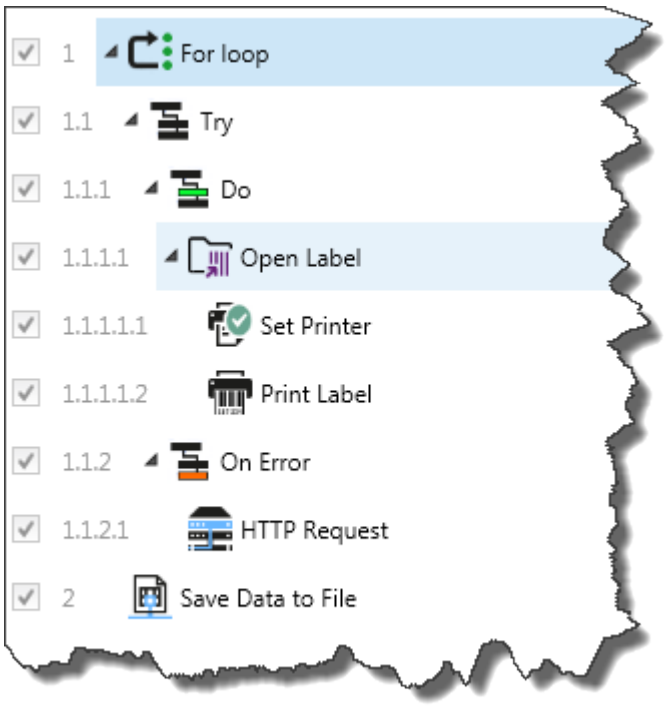
- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

For Loop (bouclage)

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Enterprise**.

Exécute plusieurs fois les actions définies en-dessous de cette action. Vous utiliserez cette action quand vous voulez exécuter plusieurs fois un groupe d'actions imbriquées. Toutes les actions

imbriquées seront exécutées en boucle autant de fois que ce qui est défini par la différence entre les valeurs de départ et de fin.



Paramètres de bouclage

- **Valeur de départ.** Spécifie la référence pour le point de départ. Vous pouvez utiliser une valeur négative. L'option **Variable** active la valeur de départ de la variable. Vous devez sélectionner une variable qui contiendra une valeur numérique pour le départ.
- **Valeur de fin.** Spécifie la référence pour le point final. Vous pouvez utiliser une valeur négative. L'option **Variable** active la valeur finale de la variable. Vous devez sélectionner une variable qui contiendra une valeur numérique pour la fin.
- **Sauvegarder la variable de bouclage vers une variable.** Sauve le pas de bouclage actuel dans une variable sélectionnée. La valeur du pas de bouclage peut contenir n'importe quelle valeur entre la valeur de départ et de fin. Vous sauvegarderez la valeur vers une variable pour l'utiliser dans une autre action et connaître l'itération actuelle.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

EXEMPLE: A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

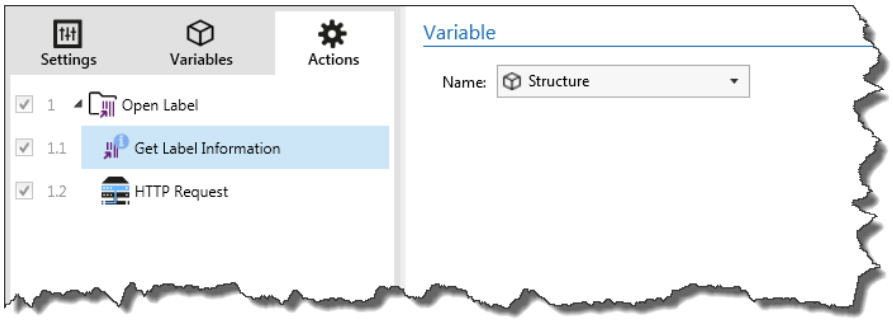
Obtenir Les Informations De L'étiquette

Cette action renvoie les informations structurelles du fichier d'étiquette associé. L'action fournit les informations concernant les dimensions de l'étiquette, le pilote d'imprimante et la liste de toutes les variables avec leurs caractéristiques principales. Elle renvoie les informations originales telles qu'elles sont sauvegardées dans le fichier d'étiquette ainsi que les informations après la fin de simulation du processus d'impression.

La simulation assure que toutes les variables d'étiquette reçoivent les valeurs comme c'est le cas durant une impression normale. Également les informations de hauteur de l'étiquette pour contrôler les dimensions correctes dans le cas où la hauteur est fournie comme variable (dans ce cas, les dimensions de l'étiquette dépendent de la quantité de données à imprimer).

L'action renverra les dimensions pour une dimension d'étiquette, pas pour une dimension de page.

Cette action sauvegardera l'information de structure de l'étiquette dans la variable sélectionnée. Vous pouvez ensuite renvoyer les données au système qui utilise l'action Requête HTTP (ou une action de connexion externe similaire), ou les renvoyer dans la réponse du déclencheur, si c'est un déclencheur bidirectionnel.



NOTE: Cette action doit être imbriquée sous l'action Open Label - Ouvrir l'étiquette.

Variable

- **Nom.** Spécifie le nom de la variable. Vous devez sélectionner une variable dans laquelle les informations de l'étiquette sont sauvegardées en format XML.
    - Si vous voulez utiliser les informations de l'XML dans le déclencheur, vous pouvez définir le Configuration de filtre XML et l'exécuter avec l'action Utilisation d'un Filtre de Données (utiliser un filtre de données).

- Si vous voulez renvoyer les données XML comme réponse dans votre déclencheur HTTP ou Web Service, utilisez cette variable directement dans le champ **Response data** (donnée de réponse) dans la configuration du déclencheur.
- Si vous voulez sauvegarder les données XML dans un fichier, utilisez l'action Sauvegarder les données dans un fichier (sauvegarder les données dans le fichier).

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  > **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Informations XML de l'échantillon d'étiquette

L'échantillon présente une vue structurelle des éléments et leurs attributs tels qu'ils sont renvoyés.

```
<?xml version="1.0" encoding="UTF-8"?>
<Label>
<Original>
. <Width>25000</Width>
<Height>179670</Height>
<PrinterName>QLS 3001 Xe</Printer>
</Original>
<Current>
<Width>25000</Width>
<Height>15120</Height>
<PrinterName>QLS 3001 Xe</Printer>
</Current>
<Variables>
<Variable>
<Name>barcode</Name>
<Description></Description>
<DefaultValue></DefaultValue>
<Format>All</Format>
<CurrentValue></CurrentValue>
<IncrementType>None</IncrementType>
<IncrementStep>0</IncrementStep>
```

```
<IncrementCount>0</IncrementCount>
<Length>100</Length>
</Variable>
</Variables>
</Format>
```

Spécification XML d'Information de l'étiquette

Cette section contient la description de structure du fichier XML telle qu'elle est renvoyée par cette action.

NOTE: Toutes les valeurs de mesures sont exprimées en unités 1/1000 mm. Par exemple, une largeur de 25000 est 25 mm.

- **<Label>.** Ceci est un élément de base.
- **<Original>.** Spécifie les dimensions de l'étiquette et le nom d'imprimante tels qu'ils sont sauvegardés dans le fichier d'étiquette.
    - **Width.** Cet élément contient la largeur originale de l'étiquette.
    - **Height.** Cet élément contient la hauteur originale de l'étiquette.
    - **PrinterName.** Cet élément contient le nom de l'imprimante pour laquelle l'étiquette a été créée.
- **Actuel.** Spécifie les dimensions de l'étiquette et le nom de l'imprimante après que la simulation d'impression a été terminée.
    - **Width.** Cet élément contient la largeur actuelle de l'étiquette.
    - **Height.** Cet élément contient la hauteur actuelle de l'étiquette. Si l'étiquette a une hauteur définie par variable, elle peut augmenter avec l'objet d'étiquette. Par exemple, les objets Text Box et RTF peuvent grandir dans le sens vertical et causer l'expansion de l'étiquette.
    - **PrinterName.** Cet élément contient le nom de l'imprimante qui sera utilisée pour l'impression. Par exemple, l'imprimante sera différente du nom d'imprimante original, si l'imprimante originale n'est pas installée sur cet ordinateur, ou si vous avez changé l'imprimante par l'action Définir l'imprimante.
- **<Variables> et <Variable>.** L'élément `Variables` contient la liste de toutes les variables d'étiquette à invitation, chacune définie dans une élément `Variable` séparé. Les variables à invitation sont celles qui sont listées dans l'interface de dialogue quand vous imprimez l'étiquette depuis le développeur. S'il n'y a pas de variables à invitation dans l'étiquette, l'élément `Variables` est vide.
    - **Nom.** Contient le nom de variable.
    - **Description.** Contient la description de la variable.
    - **DefaultValue.** Contient les valeurs par défaut telles qu'elles ont été définies pour la variable lors de la création de l'étiquette.
    - **Format.** Contient le type de contenu (caractères) que la variable accepte.
    - **CurrentValue.** Contient la valeur actuelle telle qu'elle sera utilisée pour l'impression.

- **IncrementType.** Contient l'information, si la variable est définie comme un compteur et, si c'est le cas, du type de compteur.
- **IncrementStep.** Contient l'information concernant le pas incrémental. La valeur du compteur incrémente/décrémente de cette valeur sur l'étiquette suivante.
- **IncrementCount.** Contient l'information du moment où le compteur doit incrémenter/décrémenter sa valeur. Généralement, le compteur change de valeur à chaque étiquette, mais cela peut être modifié.
- **Lenght.** Contient le nombre maximum de caractères que la variable peut contenir.

Définition du Schéma XML (XSD) pour la Spécification XML de l'étiquette

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="Format" xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="Label">
<xs:complexType>
<xs:all>
<xs:element name="Original">
<xs:complexType>
<xs:sequence>
<xs:element name="Width" type="xs:decimal" minOccurs="1" />
<xs:element name="Height" type="xs:decimal" minOccurs="1" />
<xs:element name="PrinterName" type="xs:string" minOccurs="1" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Current">
<xs:complexType>
<xs:sequence>
<xs:element name="Width" type="xs:decimal" minOccurs="1" />
<xs:element name="Height" type="xs:decimal" minOccurs="1" />
<xs:element name="PrinterName" type="xs:string" minOccurs="1" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Variables">
<xs:complexType>
<xs:sequence>
<xs:element name="Variable" minOccurs="0" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="Name" type="xs:string" minOccurs="1" />
<xs:element name="Description" type="xs:string" minOccurs="1" />
<xs:element name="DefaultValue" type="xs:string" minOccurs="1" />
<xs:element name="Format" type="xs:string" minOccurs="1" />
<xs:element name="CurrentValue" type="xs:string" minOccurs="1" />
<xs:element name="IncrementType" type="xs:string" minOccurs="1" />
<xs:element name="IncrementStep" type="xs:integer" minOccurs="1" />
<xs:element name="IncrementCount" type="xs:integer" minOccurs="1" />
<xs:element name="Length" type="xs:string" minOccurs="1" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:all>
</xs:complexType>
</xs:element>
```

```
</xs:schema>
```

Requête HTTP

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Enterprise**.

Envoie les données au serveur Web de destination en utilisant la méthode HTTP sélectionnée. Vous pouvez utiliser les schémas URI http et HTTPS.

HTTP fonctionne comme un protocole de requête de réponse dans le modèle informatique client-serveur. Par cette action, NiceLabel Automation prend le rôle de client, communiquant avec le serveur distant. Cette action va soumettre la requête HTTP sélectionnée au serveur. Le serveur renverra un message de réponse, qui peut contenir les informations du statut de complétion de la requête et peut aussi contenir le contenu requis dans le corps du message.

Paramètres de Connexion

NOTE: Cette action supporte le Protocole Internet version 6 (IPv6).

- **Destination.** Définit l'adresse, le port et la destination (chemin) sur le serveur Web. Si le serveur Web fonctionne sur le port 80 par défaut, vous pouvez sauter le numéro de port. Vous pouvez coder-en-dur les paramètres de connexion et utiliser un nom d'hôte et une adresse IP fixes. Vous pouvez également utiliser des paramètres de connexion variables. Pour plus d'informations, consultez l'article Utiliser des valeurs composées.

EXEMPLE: Si la variable `hostname` procure le nom du serveur Web et la variable `port` fournit le numéro de port, vous pouvez entre la destination suivante :
`[hostname]:[port] (nom d'hôte)`

- **Request method. (méthode de requête)** Liste les méthodes disponibles pour la requête. Vous avez le choix entre POST, GET, PUT et DELETE.
- **Timeout. (délai d'expiration)** Définit le délai durant lequel le serveur essaiera d'établir la connexion.
- **Wait for status reply. (attendre la réponse d'état)** Spécifie que vous voulez recevoir un retour d'état, si les données ont bien été transmises. Le code d'état HTTP renvoyé par le serveur Web sera sauvegardé dans la variable sélectionnée. Les codes d'état dans la plage de 2XX sont des codes de succès, la réponse habituelle "OK" est le code 200. Les codes 5XX sont des erreurs du serveur.
- **Save status reply in a variable. (sauver la réponse d'état dans une variable)** Définit la variable qui sauvegardera le code d'état renvoyé par le serveur.

Authentification

Cette section vous permet d'introduire les infos d'identification nécessaires à la connexion au serveur Web. Vous devez introduire le nom et mot de passe, qui peuvent être fixes ou fournis avec la valeur de la variable.

L'authentification Basique HTTP (BA) utilise les en-têtes statiques standard HTTP. Le mécanisme BA ne procure aucune protection de confidentialité pour les infos d'identification transmises. Elles sont seulement encodées en Base64 au transit, mais pas encodées ou hachées du tout. L'Authentification de Base devrait être utilisée en HTTPS.

Contenu

Cette section vous permet de définir le contenu que vous voulez envoyer au serveur Web. Vous pouvez utiliser le contenu fixe, un mixte de contenu fixe et variable, ou contenu variable seul. Pour insérer un contenu de variable, cliquez le bouton avec une flèche à droite de la zone de données et insérez la variable de la liste. Pour plus d'informations, consultez l'article Utiliser des valeurs composées.

- **Données.** Spécifie le contenu qui sera envoyé en sortie.
- **Encoding. (encodage)** Spécifie l'encodage des données.
- **Type.** Spécifie la caractéristique Content-Type pour le message HTTP. Si vous ne sélectionnez aucun type, la valeur par défaut `application/x-www-form-urlencoded` sera utilisée. Si vous ne voyez pas le type approprié dans la liste, introduisez-le vous-même.

Entêtes HTTP additionnelles

Certains serveurs HTTP (spécialement pour les services REST) requièrent l'inclusion de l'entête HTTP personnalisée dans le message. Cette section vous permet de fournir les entêtes HTTP dont vous avez besoin.

L'entête HTTP doit être introduite en respectant la syntaxe suivante :

```
header field name: header field value
```

Par exemple, pour utiliser les noms de champ `Accept`, `User-Agent` et `Content-Type`, vous pouvez utiliser la suivante :

```
Accept: application/json; charset=utf-8
User-Agent: User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/31.0.1650.63 Safari/537.36
Content-Type: application/json; charset=UTF-8
```

Vous pouvez coder-en-dur les noms des champs d'entête, ou vous pouvez obtenir leurs valeurs par les variables des déclencheurs. Pour accéder aux variables, cliquez la petite flèche à la droite du champ de texte. Pour plus d'informations, consultez l'article Utiliser des valeurs composées.

Vous pouvez utiliser autant de champs d'entête personnalisés que vous voulez, assurez-vous seulement que chaque entête est placée sur une nouvelle ligne.

NOTE: Les entêtes HTTP introduites vont substituer les entêtes qui sont déjà définies dans les caractéristiques des actions, telles que **Content-Type** (type de contenu).

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action

actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

> **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Charger Les Données Variables

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Charge les valeurs d'une ou plusieurs variables du fichier associé, comme sauvegardées dans le fichier par l'action **Save Variable Data**. (sauvegarder la donnée de variable) Cette action vous permet d'échanger des données entre les déclencheurs. Vous pouvez charger une variable particulière ou toutes les variables qui existent dans le fichier.

Fichier

- **File name.** (nom de fichier) Spécifie le nom de fichier duquel les valeurs variable seront chargées. Il peut être codé-en-dur, et les valeurs seront chargées chaque fois du même fichier. L'option **Variable** active le nom de fichier variable. Vous devez sélectionner une variable qui contient le chemin et/ou le nom du fichier quand le déclencheur est exécuté. En général, la valeur de la variable est assignée par un filtre.
  Utilisez la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

Structure du Fichier

Cette section définit la structure du fichier variable. La structure doit correspondre à la structure qui a été utilisée pour la sauvegarde des variables dans le fichier.

- **Delimiter.** (séparateur) Spécifie le séparateur dans le fichier de données. Vous pouvez sélectionner un séparateur prédéfini, ou introduire un personnalisé.
- **Text qualifier.** (qualificateur de texte) Spécifie le qualificateur du fichier texte. Vous pouvez sélectionner un délimiteur prédéfini, ou introduire un personnalisé.
- **Encoding.** (encodage) Spécifie le mode d'encodage utilisé dans le fichier de données. UTF-8 est une bonne sélection par défaut.

Variables

Cette section définit les variables qui seront lues dans le fichier de données. Les valeurs des variables existantes seront remplacées par les valeurs du fichier.

- **All variables.** (toutes les variables) Spécifie que toutes les variables définies dans le fichier de données seront lues.

- **Selected variables.** (variables sélectionnées) Spécifie que seules les variables sélectionnées seront lue dans le fichier de données.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  EXEMPLE: A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

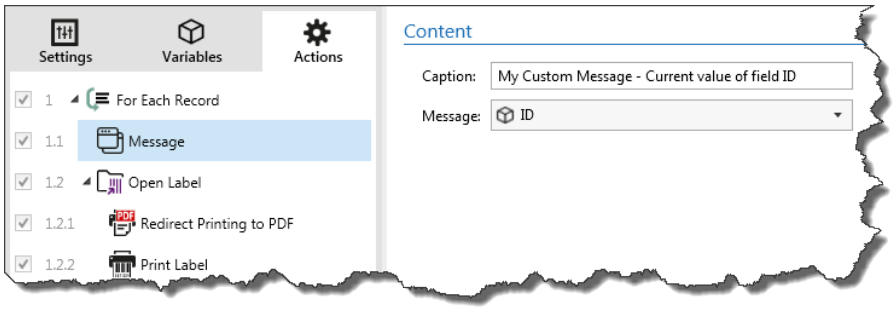
- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

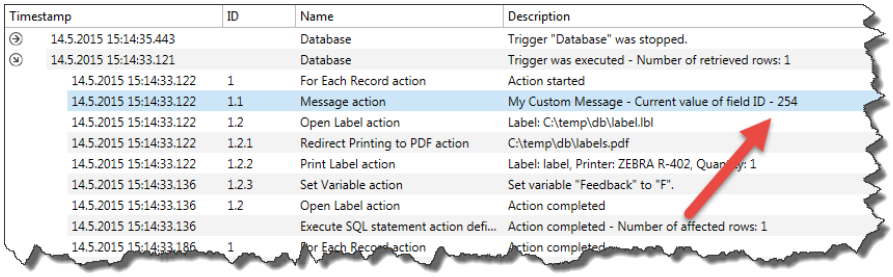
Message

Écrit un enregistrement personnalisé dans le journal.

Généralement, le fichier contient des chaînes de caractères et descriptions d'erreurs générées par l'application. Utilisez cette action pour écrire des chaînes de caractères personnalisées. C'est utile pour la résolution de problèmes et erreurs de configuration, vous permettant de surveiller les valeurs de variables sélectionnées.

EXEMPLE: Pour configurer la journalisation des messages personnalisés dans le journal de Automation Builder (quand vous testez la configuration) ou dans le journal de Automation Manager (quand le déclencheur a été implémenté et démarré), consultez les copies d'écran suivantes.





Timestamp	ID	Name	Description
14.5.2015 15:14:35.443		Database	Trigger "Database" was stopped.
14.5.2015 15:14:33.121		Database	Trigger was executed - Number of retrieved rows: 1
14.5.2015 15:14:33.122	1	For Each Record action	Action started
14.5.2015 15:14:33.122	1.1	Message action	My Custom Message - Current value of field ID - 254
14.5.2015 15:14:33.122	1.2	Open Label action	Label: C:\temp\db\label.lbl
14.5.2015 15:14:33.122	1.2.1	Redirect Printing to PDF action	C:\temp\db\labels.pdf
14.5.2015 15:14:33.122	1.2.2	Print Label action	Label: label, Printer: ZEBRA R-402, Quan[illegible]y: 1
14.5.2015 15:14:33.136	1.2.3	Set Variable action	Set variable "Feedback" to "F".
14.5.2015 15:14:33.136	1.2	Open Label action	Action completed
14.5.2015 15:14:33.136		Execute SQL statement action defi...	Action completed - Number of affected rows: 1
14.5.2015 15:14:33.186	1	For Each Record action	Action completed

Contenus

- **Caption. (légende)** Spécifie le titre du message personnalisé. L'option **Variable** active le titre de variable. Vous devez sélectionner une variable qui contient le titre quand le déclencheur est exécuté.
- **Message** Spécifie le contenu du message personnalisé. L'option **Variable** active le titre de variable. Vous devez sélectionner une variable qui contient le titre quand le déclencheur est exécuté. Généralement, vous préparerez des contenus de variables dans une autre action, ensuite les utiliser ici.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  > **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

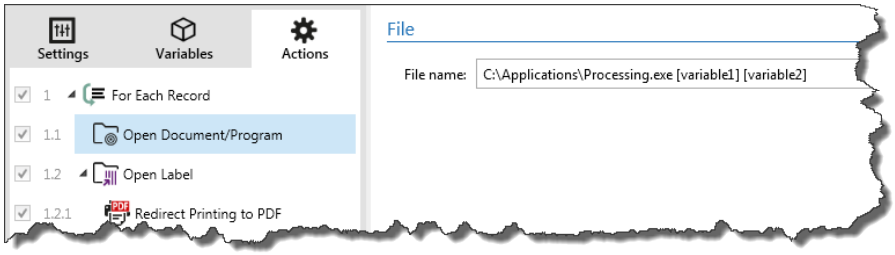
- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Ouvrir Un Document / Programme

Il interface avec le programme externe et l'exécute dans la ligne de commande. Le programme externe peut exécuter des fonctions additionnelles et renvoyer le résultat au NiceLabel Automation. Cette action permet au NiceLabel Automation de se lier avec un logiciel tierce qui peut exécuter le traitement de données additionnelles, ou acquérir des données. Le logiciel externe peut fournir des réponses de données en les sauvegardant dans un fichier, duquel vous pouvez les transposer vers des variables.

Vous pouvez fournir les valeurs de variable(s) au programme en les entourant de crochets dans la ligne de commande.

```
C:\Applications\Processing.exe [variable1] [variable2]
```



Fichier

- **Nom de fichier.** Spécifie le chemin et nom du fichier. Il peut être codé-en-dur, et le même fichier sera utilisé à chaque fois. Si vous n'utilisez que le nom de fichier sans le chemin, le dossier où est sauvegardé le fichier de configuration (.MISX) sera utilisé. Vous pouvez utiliser une référence relative au nom de fichier, où le dossier avec le fichier .MISX est utilisé comme dossier racine. L'option **Variable** active le nom de fichier variable. Vous pouvez sélectionner une variable simple qui contient le chemin et/ou le nom de fichier ou vous pouvez combiner différentes variables qui vont créer le nom de fichier. Pour plus d'informations, consultez l'article Utiliser des valeurs composées.

  NOTE: Utilisez la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

Options d'Exécution

- **Hide window. (cacher la fenêtre)** Spécifie que la fenêtre d'application ne sera pas montée à l'utilisateur. Comme NiceLabel Automation fonctionne comme une application de service dans sa propre session, il ne peut pas inter-acter avec le bureau de l'utilisateur, même s'il fonctionne avec les privilèges de l'utilisateur actuellement connecté. Pour des raisons de sécurité, Microsoft empêche cette interaction dans Windows Vista et les systèmes d'exploitation plus récents.
- **Wait for completion. (attendre l'achèvement)** Spécifie que l'exécution de l'action attendra la fin de cette action avant de continuer l'action suivante dans la liste. Activez cette option si l'action suivante dépend du résultat de l'application externe.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas

l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

> **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

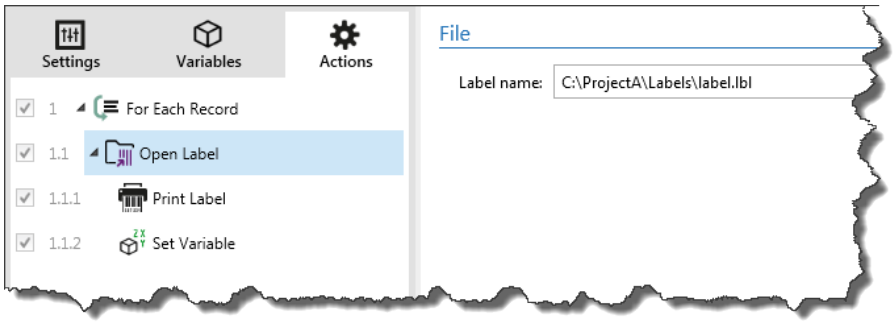
- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Open Label - Ouvrir L'étiquette

Spécifie le nom du fichier d'étiquette à imprimer. Quand l'action est exécutée, le modèle d'étiquette spécifié s'ouvre en mémoire cache. L'étiquette reste en cache tant que le déclencheur l'utilise. Le nombre d'étiquettes pouvant être ouvertes en même temps n'est pas limité. Si l'étiquette est déjà chargée et est à nouveau demandée, NiceLabel Automation va d'abord déterminer si une version plus récente est disponible et approuvée pour l'impression, ensuite va l'ouvrir.

Dans cet exemple, le NiceLabel Automation va charger l'étiquette `label.lbl` du dossier `C:\ProjectA\Labels`.

```
C:\ProjectA\Labels\label.lbl
```



Si l'étiquette spécifiée ne peut pas être trouvée, NiceLabel Automation va essayer de la trouver dans les emplacements alternatif. Pour plus d'informations, consultez l'article Ordre de recherche pour les fichiers requis.

Fichier

- **Étiquette.** Spécifie le nom d'étiquette. Il peut être codé-en-dur, et la même étiquette sera imprimée à chaque fois. L'option **Variable** active le nom de fichier variable. Vous devez sélectionner une variable qui contient le chemin et/ou le nom du fichier quand le déclencheur est exécuté. En général, la valeur vers la variable est assignée par un filtre.
  Utilisez la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

Utilisation de Chemins Relatifs

Vous pouvez également utiliser des chemins relatifs pour référencer vos fichiers d'étiquettes. Le dossier de configuration (MISX) est toujours stocké dans le dossier racine.

En utilisant la syntaxe suivante, l'étiquette se chargera de façon relative depuis l'emplacement du fichier de configuration. L'étiquette sera recherchée dans le dossier `ProjectA`, ce qui est à deux niveaux plus haut que le dossier actuel, ensuite dans le dossier `Labels`.

```
..\..\ProjectA\Labels\label.lbl
```

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  > **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Preview Label - Aperçu D'étiquette

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Enterprise**.

Exécute le processus d'impression et fournit un aperçu de l'étiquette sous forme d'image. Par défaut, l'aperçu est sauvegardé sur le disque en format image JPEG, mais vous pouvez choisir un autre type d'image. Vous pouvez également contrôler la taille de l'image d'aperçu créée image. L'action va générer l'aperçu pour une étiquette.

Dès que vous avez créé l'aperçu d'étiquette dans un fichier, vous pouvez envoyer le fichier à une application tierce en utilisant une des actions de sortie, telle que Send Data to HTTP (envoyer les données à HTTP), Send Data to Serial Port (envoyer les données au port série), Send Data to TCP/IP Port (envoyer les données au port TCP/IP), ou l'utiliser comme message de réponse des déclencheurs bidirectionnels, tels que Déclencheur de Serveur HTTP (déclencheur serveur HTTP) et Web Service Trigger (déclencheur Web Service). L'application tierce peut prendre l'image et la montrer comme aperçu à l'utilisateur.

Aperçu

- **Nom de fichier.** Spécifie le chemin et nom du fichier. Il peut être codé-en-dur, et le même fichier sera utilisé à chaque fois. Si vous n'utilisez que le nom de fichier sans le chemin, le dossier où est sauvegardé le fichier de configuration (.MISX) sera utilisé. Vous pouvez utiliser une

référence relative au nom de fichier, où le dossier avec le fichier .MISX est utilisé comme dossier racine. L'option **Variable** active le nom de fichier variable. Vous devez sélectionner une variable qui contient le chemin et/ou le nom du fichier quand le déclencheur est exécuté. En général, la valeur vers la variable est assignée par un filtre.

- **Type d'image.** Spécifie le type d'image dans lequel l'aperçu de l'étiquette sera sauvegardé.
- **Aperçu de la face arrière de l'étiquette (étiquettes double-face).** Active l'aperçu de l'arrière de l'étiquette. C'est utile quand vous utilisez des étiquettes double-face et désirez voir l'aperçu de la face arrière.

Aperçu de dimension

- **Largeur et hauteur.** Spécifie les dimensions de l'image en pixels, telle qu'elle sera sauvegardée sur le disque. L'aperçu de l'étiquette correspondra aux dimensions définies et ne sera jamais étiré. L'image utilisera un fond blanc pour tous les excès de fond qui ne sont pas couverts par l'aperçu d'étiquette.
- **Utiliser les paramètres de l'imprimante et de l'étiquette pour déterminer la taille de l'aperçu.** Active l'utilisation des dimensions exactes pour l'aperçu d'image (en pixels), comme défini dans le modèle d'étiquette (fichier .LBL) et dans les caractéristiques de l'imprimante. Le modèle d'étiquette fournira les dimensions de l'étiquette (dans l'unité de mesure sélectionnée) et le pilote d'imprimante fournira la résolution de l'imprimante (DPI). Si voulez créer un aperçu d'étiquette avec une résolution différente, changez d'imprimante avant d'exécuter l'action d'aperçu d'étiquette - Preview Label. Utilisez l'action Définir l'imprimante pour changer l'imprimante associée.

> **EXEMPLE:** Par exemple, si votre modèle d'étiquette définit les dimension de 4" × 3" et l'imprimante d'étiquettes a une résolution de 200 DPI, l'aperçu d'image résultant aura les dimensions de 800 × 600 pixels. La largeur égale à 4 pouces multiplié par 200 DPI, donne un résultat de 800 pixels. La hauteur égale à 3 pouces multiplié par 200 DPI, donne un résultat de 600 pixels.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

> **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Impression D'étiquette

Exécute l'impression de l'étiquette. Cette actions ne peut pas être utilisée seule. Vous devez toujours imbriquer cette action sous l'action Open Label - Ouvrir l'étiquette (ouvrir l'étiquette) pour référencer à l'étiquette à imprimer. Ceci permet d'avoir plusieurs étiquettes ouvertes en même temps, et vous pouvez spécifier l'étiquette à imprimer. Quand vous utilisez cette commande, l'étiquette sera imprimée en utilisant les pilotes définis dans le modèle d'étiquette. Si ce pilote d'imprimante n'est pas trouvé dans le système, l'étiquette sera imprimée en utilisant le pilote d'impression par défaut. Vous pouvez substituer le pilote d'imprimante en utilisant la commande Définir l'imprimante (définir l'imprimante).

Pour atteindre une haute performance d'impression d'étiquettes, NiceLabel Automation active deux paramètres par défaut :

- **Processus parallèle.** De multiples processus d'impression sont effectués simultanément. Le nombre de tâches d'impression fonctionnant en arrière-plan dépend du matériel, spécifiquement du type de processeur. Chaque noyau de processeur peut accommoder une tâche d'impression, et cette valeur par défaut peut être changée. Pour plus d'informations, voir l'article Changer les paramètres par défaut d'Impressions Multiples.
- **Mode asynchrone.** Dès que le pré-traitement du déclencheur se termine et les instructions pour l'engin d'impression sont disponibles, la tâche d'impression démarre en arrière-plan. Le contrôle est rendu au déclencheur pour qu'il puisse accepter le flux de données entrantes suivant dès que possible. Quand le mode synchrone est activé, le contrôle n'est pas rendu au déclencheur tant que le processus d'impression n'est pas terminé. Cela peut prendre un certain temps, mais le déclencheur a le bénéfice de pouvoir envoyer un retour d'infos à l'application fournissant les données. Pour plus d'informations, consultez l'article Mode d'impression Synchrone.

  NOTE: L'utilisation de l'option **Sauvegarder l'erreur dans une variable**, dans Exécution d'action et Traitement d'Erreurs ne va pas produire de résultats en mode asynchrone, car le déclencheur ne recevra pas de retour du processus d'impression. Pour capturer un retour du processus d'impression, vous devez activer le mode synchrone.

Quantité

Cette section spécifie le nombre d'étiquettes que vous voulez imprimer.

- **Étiquettes.** Spécifie le nombre d'étiquettes à imprimer.
- **Variable** Spécifie la variable qui va définir la quantité d'étiquettes. La valeur de variable est généralement assignée par l'action **Utiliser un Filtre de Données** et doit être intègre.
- **Tout (quantité illimitée).** Selon le design du modèle d'étiquette, les étiquettes seront imprimées en qualités différentes.

  Détails

  Typiquement, cette option est utilisée dans deux scénarios.

1. Commander à l'imprimante d'imprimer la même étiquette continuellement jusqu'à ce qu'elle soit coupée, ou qu'elle reçoive une commande d'effacement de sa mémoire tampon.

   AVERTISSEMENT Dans ce scénario, vous devez utiliser le pilote d'imprimante NiceLabel pour imprimer vos étiquettes.

   Quand vous imprimez une étiquette fixe, une seule tâche d'impression est envoyée à l'imprimante, avec la quantité définie à "illimitée". Les imprimantes d'étiquettes ont un paramètre de la commande d'impression pour indiquer l'impression "illimitée".

   Quand l'étiquette n'est pas fixe mais inclut des objets qui changent durant l'impression, comme des compteurs, alors la quantité imprimée sera définie par la quantité maximale supportée par l'imprimante. Le pilote d'imprimante NiceLabel connait la quantité limite et imprimera cette quantité d'étiquettes.

   EXEMPLE: La quantité maximale supportée par l'imprimante est de 32.000. C'est la quantité d'étiquettes qui sera imprimée si vous sélectionnez la quantité "illimitée".

2. Le déclencheur ne fournit aucune donnée, mais sert seulement de signal que "l'évènement s'est effectué". La logique pour acquérir les données nécessaires sur l'étiquette. Généralement, une connexion vers une base de données est configurée avec l'étiquette et, à chaque déclenchement, l'étiquette doit se connecter à la base de données et charger les données. Dans ce cas, l'option "illimitée" se comprend comme "imprimer toutes les données de la base de données".

- **Quantité variable (définie par la variable d'étiquette).** Spécifie qu'une certaine variable d'étiquette contient l'information de quantité d'étiquettes. Le déclencheur ne reçoit pas le nombre d'étiquettes à imprimer, donc il transmet la décision au modèle d'étiquette. L'étiquette peut contenir une connexion à la base de données, qui fournira la quantité d'étiquettes, ou il y a une autre source d'information de quantité. Une variable d'étiquette doit être définie comme "variable de quantité". pour plus d'informations, consultez le guide d'utilisation du concepteur d'étiquettes.

Avancé

Cette section spécifie les paramètres de quantité liés aux quantités d'étiquettes moins fréquemment utilisés.

- **Nombre d'étiquettes sautées.** Spécifie le nombre d'étiquettes qui seront sautées à la première page d'étiquettes. La feuille d'étiquettes peut avoir déjà été imprimée, mais pas entièrement. Vous pouvez utiliser la même feuille, en déplaçant la position de départ. Cette option est applicable, quand vous imprimez les étiquettes sur des feuilles, pas des rouleaux d'étiquettes, c'est donc applicable aux imprimantes de bureau pas pour les imprimantes d'étiquettes. La valeur peut être codée-en-dur, ou une variable peut fournir le nombre.
- **Copies Identiques d'étiquettes** Spécifie le nombre de copies d'étiquettes à réaliser pour chaque étiquette unique. Cette option produit le même résultat que le l'option principale Nombre d'Étiquettes, quand vous avez des étiquettes fixes. Avec des étiquettes variables, telles que les étiquettes utilisant des compteurs, vous pouvez obtenir le nombre réel de copies.

- **Ensembles d'étiquettes.** Spécifie combien de fois le processus complet d'impression doit être répété. Par exemple, le déclencheur reçoit un contenu de 3 lignes de données formatées CSV, donc on attend l'impression de 3 étiquettes (1, 2, 3). Si vous mettez cette option à 3, l'impression se fera dans l'ordre suivant : 1, 2, 3, 1, 2, 3, 1, 2, 3.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  > **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

État De L'imprimante

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Communique avec l'imprimante pour obtenir sont état en temps réel. Le résultat est l'information concernant les erreurs, l'état de la file d'attente, le nombre de tâches dans la file d'attente est collecté, ce qui vous permet d'identifier les erreurs possibles. Il y a des conditions préalables pour activer le rapportage d'état de l'imprimante :

- Vous devez utiliser Pilote d'Imprimantes NiceLabel pour recevoir les informations d'état détaillées.
- L'imprimante doit être capable de rapporter l'état en temps réel.
- L'interface que vous utilisez pour vous connecter à l'imprimante doit supporter la messagerie bidirectionnelle.

Exemples d'utilisation possible. (1) Vous allez vérifier l'état de l'imprimante avant l'impression. Si l'imprimante est en état d'erreur, vous allez imprimer l'étiquette sur l'imprimante de réserve. (2) Vous allez compter le nombre de tâches en attente dans le spouleur de l'imprimante principale.

S'il y en a de trop, vous allez imprimer sur une imprimante alternative. (3) Vous allez vérifier l'état de l'imprimante avant l'impression. Si l'imprimante est en état d'erreur, vous n'allez pas imprimer l'étiquette, mais rapporter l'erreur au système principal en utilisant une des actions de sortie, telles que Envoyer les données au port TCP/IP, Requête HTTP, Exécuter une instruction SQL, Service Web ou comme réponse du déclencheur.

Imprimante

- **Nom d'imprimante.** Spécifie le nom d'imprimante. Vous pouvez sélectionner l'imprimante dans la liste d'imprimantes installées localement, ou vous pouvez entrer un nom d'imprimante. L'option **Variable** active l'imprimante variable. Quand activée, vous devez sélectionner une variable qui contiendra le nom d'imprimante quand le déclencheur est exécuté. En général, la valeur vers la variable est assignée par un filtre.

Mappage de données

AVERTISSEMENT La majorité des paramètres suivants est seulement supportée avec l'utilisation de Pilote d'Imprimantes NiceLabel. Si vous utilisez un autre pilote d'imprimante, vous ne pouvez utiliser que les paramètres relatifs au spouleur.

Cette section définit les paramètres qui sont renvoyés comme résultat de l'action État d'Imprimante.

- **État d'imprimante.** Spécifie l'état de l'imprimante en temps réel comme une chaîne de caractères. Si l'imprimante est dans différents états, tous les états sont fusionnés ensemble dans une chaîne de caractères, délimités par des virgules ",". S'il n'y a pas de problèmes avec l'imprimante, ce champ n'a pas de valeur. L'état de l'imprimante peut être "Hors-ligne", "Plus d'étiquettes", "Ruban approche la fin". Il n'y a pas de rapportage standardisé, donc chaque marque d'imprimante peut utiliser des messages d'état différents.
- **Erreur imprimante.** Spécifie la valeur booléenne (vrai/faux) de l'état d'erreur de l'imprimante.
- **Imprimante hors-ligne.** Spécifie la valeur booléenne (vrai/faux) de l'état hors-ligne de l'imprimante.
- **Pilote en pause.** Spécifie la valeur booléenne (vrai/faux) de l'état de pause du pilote d'imprimante.
- Pilote d'Imprimantes NiceLabel **pilote.** Spécifie la valeur booléenne (vrai/faux) de l'état du Pilote d'Imprimantes NiceLabel. Procure l'information si le pilote sélectionné est Pilote d'Imprimantes NiceLabel.
- **État du spouleur.** Spécifie l'état du spouleur comme une chaîne de caractères, comme rapporté par le système Windows. Le spouleur peut être simultanément dans différents états. Dans ce cas, les états sont fusionnés avec des virgules ",".
- **ID d'état du spouleur.** Spécifie l'état du spouleur comme un nombre, comme rapporté par le système Windows. Le spouleur peut être simultanément dans différents états. Dans ce cas, les ID d'état contiennent toutes les ID comme indicateurs. Par exemple, la valeur 5 représente les ID des états 4 et 1, ce qui se traduit par "Imprimante en erreur, Imprimante en pause". Référez-vous au tableau ci-dessus.

L'action retournera une valeur décimale, les valeurs dans le tableau ci-dessous sont en hexadécimal, vous devrez donc les convertir avant d'analyser la réponse.

Table d'ID d'états du spouleur et descriptions correspondantes

ID d'état du spouleur (en hexadécimal)	Description de l'état du spouleur
0	Pas d'état.
1	Imprimante en pause.
2	Imprimante en impression.
4	Imprimante en erreur.
8	Imprimante pas disponible.
10	L'imprimante n'a plus de papier.
20	Alimentation manuelle requise.
40	L'imprimante a un problème de papier.
80	Imprimante hors-ligne.
100	État Entrée/Sortie actif.
200	Imprimante occupée.
400	Blocage de papier.
800	Corbeille de sortie pleine.
2000	Imprimante en attente.
4000	Imprimante en cours d'exécution.
10000	Imprimante en pré-chauffage.
20000	Niveau d'encre bas.
40000	Plus d'encre dans l'imprimante.
80000	La page actuelle ne peut pas être imprimée.
100000	Une intervention de l'utilisateur est requise.
200000	L'imprimante n'a plus de mémoire disponible.
400000	Protection ouverte.
800000	Erreur inconnue.
1000000	L'imprimante est en mode de sauvegarde d'énergie.

- **Nombre de tâches dans le spouleur.** Spécifie le nombre de tâches qui sont dans le spouleur pour l'imprimante sélectionnée.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

EXEMPLE: A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Lecture De Données Du Fichier

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Lit le contenu du nom de fichier fourni et le sauvegarde dans la variable. Vous pouvez lire le contenu de fichiers de n'importe quel type, y compris des données binaires.

Généralement NiceLabel Automation recevra les données pour l'impression de l'étiquette avec le déclencheur. Par ex. quand vous utilisez le déclencheur de fichier, le contenu du fichier de déclencheur est automatiquement lu et peut être analysé par des filtres. Vous pouvez toutefois vouloir contourner les filtres pour obtenir des données externes. Dès que vous exécutez cette action et avez sauvegardé les données dans une variable, vous pouvez utiliser une des actions disponibles pour employer les données.

Cette action est utile :

- Quand vous devez combiner les données reçues par le déclencheur avec les données sauvegardées dans un fichier.

AVERTISSEMENT Si vous chargez des données de fichiers binaires (comme des fichiers image bitmap ou d'impression), assurez-vous que la variable dans laquelle le contenu lu est sauvegardé est définie comme **variable binaire**.

- Quand vous voulez échanger des données entre les déclencheurs. Un déclencheur prépare les données et les sauvegarde dans le fichier (en utilisant l'action Sauvegarder les données dans un fichier), l'autre déclencheur lit les données.

Fichier

- **Nom de fichier.** Spécifie le chemin et nom du fichier. Il peut être codé-en-dur, et le même fichier sera utilisé à chaque fois. Si vous n'utilisez que le nom de fichier sans le chemin, le dossier où est sauvegardé le fichier de configuration (.MISX) sera utilisé. Vous pouvez utiliser une référence relative au nom de fichier, où le dossier avec le fichier .MISX est utilisé comme dossier racine. L'option **Variable** active le nom de fichier variable. Vous pouvez sélectionner une variable simple qui contient le chemin et/ou le nom de fichier ou vous pouvez combiner différentes variables qui vont créer le nom de fichier. Pour plus d'informations, consultez l'article Utiliser des valeurs composées.

NOTE: Utilisez la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

Contenu

- **Variable.** Spécifie la variable qui va sauvegarder tout le contenu du nom de fichier sélectionné. Vous devez avoir au moins une variable définie.
- **Encodage.** Spécifie l'encodage des données lues. Si vous n'êtes pas sur de l'encodage, laissez-le en **Auto**. Vous ne pouvez pas sélectionner l'encodage quand vous lisez les données dans une variable binaire. Dans ce cas, la variable contiendra les données telles qu'elles.

Ré-essai en cas d'échec

NiceLabel Automation l'accès au fichier peut être impossible car il est verrouillé par une autre application. Si une application écrit encore des données dans le fichier et l'a bloqué en mode exclusif, aucune autre application ne peut l'ouvrir en même temps, même pas en lecture. D'autres causes pour les ré-essais sont les suivantes : le fichier n'est pas (encore), le dossier n'existe pas (encore), l'utilisateur du service ne dispose pas des droits d'accès au fichier, ou autre chose a échoué.

Ces options déterminent le nombre de fois que NiceLabel Automation doit réessayer d'ouvrir un fichier. Si le fichier ne peut pas être ouvert après tous les essais, l'action va échouer.

- **Tentatives de ré-essai.** Spécifie le nombre de fois qu'on devrait réessayer d'accéder au fichier. Quand la valeur est 0, aucune tentative n'est effectuée.
- **Intervalles d'essais.** Spécifie l'intervalle de temps entre les essais (défini en millisecondes).

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  > **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Lecture Des Données Sur Le Port Série

Collecte les données reçues sur le port série (RS-232) et les sauvegarde dans une variable sélectionnée. Vous pouvez utiliser cette action pour communiquer avec les appareils externes à port série.

Port.

- **Nom de Port.** Spécifie le nom du port sur lequel se connecte votre appareil externe. Cela peut être un port COM matériel ou un port COM virtuel.

Paramètres de Port

Cette section affiche les options pour la connexion du port série. Assurez-vous que les paramètres correspondent aux paramètres de votre appareil extérieur.

- **Bits par seconde.** Spécifie la vitesse qui sera utilisée par l'appareil extérieur pour communiquer avec l'ordinateur. L'alias généralement utilisé pour ce paramètre est "baud rate".
- **Bits de données.** Spécifie le nombre de bits de données dans chaque caractère. 8 bits de données sont généralement utilisés dans les appareils récents.
- **Parité.** Spécifie la méthode de détection d'erreurs de transmission. Le paramètre de parité généralement utilisé est "aucune", avec la détection d'erreur gérée par un protocole de communication (contrôle de flux).
- **Bits d'arrêt.** Les bits d'arrêt envoyés à la fin de chaque caractère permettent à la machine de réception de détecter la fin d'un caractère et de le resynchroniser avec le flux de caractères. Les appareils électroniques utilisent généralement un bit de stop.
- **Contrôle de flux.** Un port série peut utiliser des signaux dans l'interface pour suspendre et relancer la transmission de données.

> **EXEMPLE:** Par exemple, un appareil lent peut avoir besoin d'utiliser la poignée de main avec le port série pour indiquer que les données auraient du être suspendues pendant que les processus de l'appareil reçoivent des données.

Options

- **Délai de lecture.** Spécifie le délai optionnel durant la lecture de données sur le port série. Le contenu complet du tampon de port série sera lu après le délai.
- **Envoyer les données d'initialisation.** Spécifie la chaîne de caractères qui est envoyée au port série sélectionné avant la lecture des données. Cela procure la fonctionnalité d'initialisation de l'appareil avant qu'il ne fournisse les données. Vous pouvez également l'utiliser pour envoyer une question spécifique à l'appareil, et recevoir la réponse spécifique. Cliquez les boutons à flèche pour insérer des caractères spéciaux, tels que les codes de contrôle. Pour plus d'informations, consultez l'article Introduction de Caractères Spéciaux (Codes de Contrôle).

Extraction de Données

- **Active l'extraction de données.** Procure une fonctionnalité pour extraire une partie des données. Vous pouvez définir la position de départ et de fin. Tous les caractères dans ces positions seront extraits. Pour utiliser des techniques d'extraction plus fortes, vous pouvez analyser les données reçues dans des filtres. Pour plus de renseignements, consultez l'article Comprendre les Filtres.

Résultats

- **Sauvegarder les données dans une variable.** Spécifie la variable qui va sauvegarder les données reçues. Dès que vous avez capturé les données et les avez sauvegardé dans une variable, vous pouvez les manipuler en utilisant des filtres, et/ou comme entrées pour une autre action.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront

exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.

- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  EXEMPLE: A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Rediriger L'impression Vers Un Fichier

Détourne la tâche d'impression vers un fichier. Au lieu d'envoyer le fichier d'impression créé au port d'imprimante, comme défini dans le pilote d'imprimante, l'impression est redirigée vers un fichier. Vous pouvez ajouter les données à un fichier existant ou substituer le fichier existant. Cette action permet de capturer les commandes d'impression dans un fichier. L'action redirigera seulement l'impression, assurez-vous donc qu'elle est suivie par l'action **Print Label** (imprimer l'étiquette).

NOTE: NiceLabel Automation fonctionne comme un service sous un compte Windows défini. Assurez-vous que ce compte utilisateur dispose des privilèges d'accès au fichier spécifié avec les droits de lecture/d'écriture. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

Cette action est également utile pour imprimer plusieurs étiquettes différentes (fichiers .LBL) sur l'imprimante réseau en conservant l'ordre d'étiquettes correct. Quand plusieurs fichiers .LBL sont imprimés par le même déclencheur, NiceLabel Automation va envoyer chaque étiquette dans une tâche d'impression séparée à l'imprimante, même si l'imprimante d'exécution est identique pour les deux étiquettes. Quand une imprimante réseau est utilisée, des tâches d'autres utilisateurs peuvent être insérées entre deux tâches que le déclencheur doit envoyer ensemble. L'utilisation de cette action vous permet de combiner les données d'impression dans un même fichier et ensuite d'envoyer son contenu à l'imprimante en utilisant l'action Envoyer les données à l'imprimante.

Fichier

- **File name. (nom de fichier)** Spécifie le nom de fichier. Il peut être codé-en-dur, et l'impression sera redirigée vers le même fichier à chaque fois. L'option **Variable** active le nom de fichier

variable. Vous devez sélectionner une variable qui contient le chemin et/ou le nom du fichier quand le déclencheur est exécuté. En général, la valeur vers la variable est assignée par un filtre. Utilisez la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

- **Substituer le fichier.** Si le fichier spécifié existe déjà sur le disque, il sera substitué.
- **Ajouter les données au fichier.** Le fichier de la tâche sera ajouté aux données existantes dans le fichier fourni.

Persistance

Cette option permet de contrôler la persistance de l'action de redirection. Vous pouvez contrôler le nombre d'actions Impression d'Étiquette qui sont touchées par la redirection.

- **Appliquer à toutes les actions suivantes d'impression.** Spécifie que la redirection d'impression s'applique à **toutes** les actions Impression d'Étiquette qui sont définies après cette action de redirection.
- **Appliquer à l'action d'impression suivante.** Spécifie que la redirection d'impression s'applique seulement a l'action Impression d'Étiquette **suivante** (une seule fois).

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  > **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Rediriger L'impression Vers Un PDF

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Enterprise**.

Détourne la tâche d'impression vers un document PDF. Au lieu d'imprimer l'étiquette sur l'imprimante, la tâche d'impression est détournée vers un PDF. Vous pouvez ajouter des données au fichier existant ou substituer le fichier existant. Le document PDF contiendra les dimensions exactes de l'étiquette,

comme définies durant la conception de l'étiquette. La qualité du rendu des graphiques dans les PDF correspond à la résolution de l'imprimante ciblée et des dimensions d'impression désirées.

L'action redirigera seulement l'impression, assurez-vous donc qu'elle est suivie par l'action **Print Label** (imprimer l'étiquette).

NOTE: NiceLabel Automation fonctionne comme un service sous un compte Windows défini. Assurez-vous que ce compte utilisateur dispose des privilèges d'accès au fichier spécifié avec les droits de lecture/d'écriture. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

Fichier

- **File name. (nom de fichier)** Spécifie le nom de fichier. Il peut être codé-en-dur, et l'impression sera redirigée vers le même fichier à chaque fois. L'option **Variable** active le nom de fichier variable. Vous devez sélectionner une variable qui contient le chemin et/ou le nom du fichier quand le déclencheur est exécuté. En général, la valeur vers la variable est assignée par un filtre. Utilisez la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.
- **Substituer le fichier.** Si le fichier spécifié existe déjà sur le disque, il sera substitué.
- **Ajouter les données au fichier.** Le fichier de la tâche sera ajouté aux données existantes dans le fichier fourni.

Persistance

Cette option permet de contrôler la persistance de l'action de redirection. Vous pouvez contrôler le nombre d'actions Impression d'Étiquette qui sont touchées par la redirection.

- **Appliquer à toutes les actions suivantes d'impression.** Spécifie que la redirection d'impression s'applique à **toutes** les actions Impression d'Étiquette qui sont définies après cette action de redirection.
- **Appliquer à l'action d'impression suivante.** Spécifie que la redirection d'impression s'applique seulement a l'action Impression d'Étiquette **suivante** (une seule fois).

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  EXEMPLE: A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour

effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Lancer Le Fichier Commande

Exécute les commandes du fichier de commande sélectionné. Tous les types de fichiers fournissent des commandes que NiceLabel Automation va exécuter dans l'ordre de haut en bas. Les fichiers de commande fournissent généralement des données pour une seule étiquette, mais vous pouvez définir des fichiers à niveaux de complexité divers. Pour plus d'informations, voir l'article Référence et Résolution de Problèmes.

NOTE: Cette action est disponible dans tous les produits NiceLabel Automation au niveau des composants du produit sélectionné. Au plus le niveau du produit est élevé, au plus de commandes pourront être utilisées dans les fichiers de commande.

Fichier

- **Type de fichier.** Spécifie le type de fichier de commande à exécuter.
- **Nom de fichier.** Spécifie le nom du fichier de commande. Il peut être codé-en-dur, et le même fichier de commande sera utilisé à chaque fois. L'option **Variable** active le nom de fichier variable. Vous devez sélectionner une variable qui contient le chemin et/ou le nom du fichier quand le déclencheur est exécuté. En général, la valeur de la variable est assignée par un filtre. Utilisez la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

Comment recevoir un fichier de commande dans un déclencheur et l'exécuter

Quand le déclencheur reçoit le fichier de commande et vous voulez l'exécuter, effectuer les opérations suivantes :

1. Dans l'onglet Variables, cliquez le bouton **Variable Interne** sur le ruban.
2. Dans le menu déroulant, activez la variable interne `DataFileName`. Cette variable interne fournit le nom de chemin et de fichier vers le fichier qui contient les données reçues par le déclencheur. Dans ce cas, le contenu est le fichier de commande. Pour plus d'informations, consultez l'article Variables Internes.
3. Dans l'onglet Actions, ajoutez l'action pour exécuter le fichier de commande, telle que Lancer le Fichier Commande, Utiliser un fichier de Commande Oracle XML, ou Utiliser un fichier de commande SAP AII XML.
   Pour l'action xécuter un **Fichier de Commande**, sélectionnez le type de fichier de commande dans **Type de Fichier**.
4. Activez l'option **Variable**.
5. Sélectionnez la variable `DataFileName` dans la liste de variables disponibles.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront

exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.

- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  EXEMPLE: A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Utiliser Un Fichier De Commande Oracle XML

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Exécute l'impression avec les données d'un fichier de format Oracle XML.

NiceLabel Automation supporte en interne les fichiers XML ayant la structure "Oracle XML", qui sont définis par le logiciel Oracle Warehouse Management. Utilisez cette action comme un raccourci pour exécuter les fichiers Oracle XML directement, sans devoir les analyser avec des filtres XML et mapper les valeurs dans des variables. Pour pouvoir utiliser cette action, le fichier XML doit être conforme au spécifications Oracle XML. Pour plus d'informations, voir l'article Caractéristiques Oracle XML.

Utilisez la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

Comment recevoir un fichier de commande dans un déclencheur et l'exécuter

Quand le déclencheur reçoit le fichier de commande et vous voulez l'exécuter, effectuer les opérations suivantes :

1. Dans l'onglet Variables, cliquez le bouton **Variable Interne** sur le ruban.
2. Dans le menu déroulant, activez la variable interne `DataFileName`. Cette variable interne fournit le nom de chemin et de fichier vers le fichier qui contient les données reçues par le déclencheur. Dans ce cas, le contenu est le fichier de commande. Pour plus d'informations, consultez l'article Variables Internes.
3. Dans l'onglet Actions, ajoutez l'action pour exécuter le fichier de commande, telle que Lancer le Fichier Commande, Utiliser un fichier de Commande Oracle XML, ou Utiliser un fichier de commande SAP AII XML.

Pour l'action xécuter un **Fichier de Commande**, sélectionnez le type de fichier de commande dans **Type de Fichier**.

4. Activez l'option **Variable**.
5. Sélectionnez la variable `DataFileName` dans la liste de variables disponibles.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  > **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Utiliser Un Fichier De Commande SAP AII XML

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Exécute l'impression avec les données d'un fichier de format SAP AII XML.

NiceLabel Automation supporte en interne les fichiers XML ayant la structure "SAP AII XML", qui sont définis par le logiciel SAP. Utilisez cette action comme un raccourci pour exécuter les fichiers SAP AII XML directement, sans devoir les analyser avec des filtres XML et mapper les valeurs dans des variables. Pour pouvoir utiliser cette action, le fichier XML doit être conforme au spécifications SAP AII XML. Pour plus d'informations, voir l'article Caractéristiques SAP AII XML.

Utilisez la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

Comment recevoir un fichier de commande dans un déclencheur et l'exécuter

Quand le déclencheur reçoit le fichier de commande et vous voulez l'exécuter, effectuer les opérations suivantes :

1. Dans l'onglet Variables, cliquez le bouton **Variable Interne** sur le ruban.
2. Dans le menu déroulant, activez la variable interne `DataFileName`. Cette variable interne fournit le nom de chemin et de fichier vers le fichier qui contient les données reçues par le déclencheur. Dans ce cas, le contenu est le fichier de commande. Pour plus d'informations, consultez l'article Variables Internes.
3. Dans l'onglet Actions, ajoutez l'action pour exécuter le fichier de commande, telle que Lancer le Fichier Commande, Utiliser un fichier de Commande Oracle XML, ou Utiliser un fichier de commande SAP AII XML.
   Pour l'action xécuter un **Fichier de Commande**, sélectionnez le type de fichier de commande dans **Type de Fichier**.
4. Activez l'option **Variable**.
5. Sélectionnez la variable `DataFileName` dans la liste de variables disponibles.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  > **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Sauvegarder Les Données Dans Un Fichier

> La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Sauve la valeur variable ou autres flux de données (telles que les données binaires) dans le fichier. Le service NiceLabel Automation doit avoir les droits d'accès en écriture dans le dossier spécifié.

Fichier

- **Nom de fichier.** Spécifie le chemin et nom du fichier. Il peut être codé-en-dur, et le même fichier sera utilisé à chaque fois. Si vous n'utilisez que le nom de fichier sans le chemin, le dossier

où est sauvegardé le fichier de configuration (.MISX) sera utilisé. Vous pouvez utiliser une référence relative au nom de fichier, où le dossier avec le fichier .MISX est utilisé comme dossier racine. L'option **Variable** active le nom de fichier variable. Vous pouvez sélectionner une variable simple qui contient le chemin et/ou le nom de fichier ou vous pouvez combiner différentes variables qui vont créer le nom de fichier. Pour plus d'informations, consultez l'article Utiliser des valeurs composées.

NOTE: Utilisez la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

Si le fichier existe

- **Substituer le fichier.** Indique que le fichier spécifié sera substitué s'il existe déjà sur le disque.
- **Ajouter les données au fichier.** Spécifie que les données seront écrites à la fin du fichier si le nom de fichier défini existe déjà.

Contenus

- **Utiliser les données reçues par le déclencheur.** Le fichier contiendra les données originales, comme reçues par le déclencheur. En fait, cela réalisera une copie des données entrantes.
- **Personnalisé.** Les données comprendront le contenu tel qu'il est fourni dans la zone de texte. Vous pouvez combiner des valeurs fixes, des valeurs variables et des caractères spéciaux dans le contenu. Pour insérer des variables et des caractères spéciaux, cliquez le bouton avec une flèche à droite de la zone de texte. Pour plus d'informations, consultez l'article Utiliser des valeurs composées.
- **Encodage.** Spécifie l'encodage du fichier de sortie. Sélectionnez **Auto** si vous ajoutez des données au fichier et désirez utiliser l'encodage du fichier existant.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  EXEMPLE: A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur

est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Sauvegarde Des Données Variables

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Sauvegarde les valeurs d'une ou plusieurs variables au fichier associé. Cette action vous permet d'échanger des données entre les déclencheurs. Pour relire les données dans le déclencheur, utilisez l'action **Charger les Données Variables**. Les valeurs sont sauvegardées en format CSV, avec le nom de variable sur la première ligne. Quand les variables ont des données sur plusieurs lignes, les caractères d'une nouvelle ligne (CR/LF) seront encodés comme `\n\r`.

Fichier

- **Nom de fichier.** Spécifie le nom de fichier dans lequel les valeurs variable seront chargées. Il peut être codé-en-dur, et les valeurs seront sauvegardées chaque fois dans le même fichier. L'option **Variable** active le nom de fichier variable. Vous devez sélectionner une variable qui contient le chemin et/ou le nom du fichier quand le déclencheur est exécuté. En général, la valeur de la variable est assignée par un filtre.
  Utilisez la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

Si le Fichier Existe

- **Substituer le fichier.** Spécifie que les données du fichier existant seront substitués par les nouvelles données. L'ancien contenu est perdu.
- **Ajouter les données au fichier.** Spécifie que les valeurs de variables sont ajoutées au fichier de données existant. Cette option vous permet de générer un fichier "base de donnée texte" tel qu'un fichier CSV.

Structure du Fichier

Cette section définit la structure du fichier variable. La structure doit correspondre à la structure qui a été utilisée pour la sauvegarde des variables dans le fichier.

- **Délimiteur.** Spécifie le délimiteur dans le fichier de données. Vous pouvez sélectionner un délimiteur prédéfini, ou introduire un personnalisé.
- **Text qualifier.** Spécifie le qualificateur du fichier texte. Vous pouvez sélectionner un délimiteur prédéfini, ou introduire un personnalisé.
- **Encodage.** Spécifie le mode d'encodage utilisé dans le fichier de données. UTF-8 est une bonne sélection par défaut.

Variables

Cette section définit les variables qui seront lues dans le fichier de données. Les valeurs des variables existantes seront remplacées par les valeurs du fichier.

- **Toutes variables.** Spécifie que toutes les variables définies dans le fichier de données seront lues.

- **Variables Sélectionnées.** Spécifie que seules les variables sélectionnées seront lue dans le fichier de données.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  EXEMPLE: A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Envoyer Des Commandes Personnalisées

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Exécute les commandes personnalisées introduites. Vous devez toujours imbriquer cette action sous l'action Open Label - Ouvrir l'étiquette (ouvrir l'étiquette) pour référencer l'étiquette à la commande. Pour plus d'informations, consultez l'article Commandes Personnalisées.

NOTE: La majorité des commandes personnalisées sont disponibles avec des actions individuelles, donc dans la plupart des cas vous n'avez pas besoin de commandes personnalisées.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action

actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

> **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Envoyer Les Données À L'imprimante

Envoie les données à l'imprimante sélectionnée. Utile pour envoyer des flux d'impression pré-générés à toute imprimante disponible. NiceLabel Automation utilise les pilotes d'imprimante en mode de transit, pour pouvoir envoyer les données au port de destination, tel que le port LPT, COM, TCP/IP ou USB, sur lequel l'imprimante est connectée.

> Scénario possible. Les données reçues par le déclencheur doivent être imprimées sur la même imprimante réseau mais sur d'autres modèles d'étiquettes (fichiers .LBL). L'imprimante peut accepter des données de différents postes de travail et imprimera généralement les tâches dans l'ordre reçu. NiceLabel Automation enverra chaque modèle d'étiquette dans une tâche séparée, rendant possible à d'autres postes de travail d'insérer ses tâches entre les tâches créées dans notre NiceLabel Automation. Au lieu d'envoyer chaque tâche séparément à l'imprimante, vous pouvez fusionner toutes les étiquettes (en utilisant l'action Rediriger l'impression vers un fichier et ensuite envoyer une grande tâche d'impression à l'imprimante.

Imprimante

- **Nom d'imprimante.** Spécifie le nom d'imprimante. Vous pouvez sélectionner l'imprimante dans la liste d'imprimantes installées localement, ou vous pouvez entrer un nom d'imprimante. L'option **Variable** active l'imprimante variable. Quand activée, vous devez sélectionner une variable qui contiendra le nom d'imprimante quand le déclencheur est exécuté. En général, la valeur vers la variable est assignée par un filtre.

Source de Données

Cette section vous permet de définir le contenu que vous voulez envoyer à l'imprimante.

- **Utiliser les données reçues par le déclencheur.** Définit que les données reçues par le déclencheur seront utilisées. Dans ce cas, vous recevez le flux d'impression comme entrée dans le filtre et voulez le rediriger sans aucune modification. Le même résultat peut être atteint en activant la variable interne `DataFileName` et en utilisant le contenu du fichier auquel elle se réfère. Pour plus d'informations, consultez l'article Variables Internes.
- **File name. (nom de fichier)** Définit le chemin et nom de fichier contenant le flux d'impression. Le contenu du fichier spécifié est en usage. L'option **Variable** active le nom de fichier variable. Vous devez sélectionner une variable qui contient le chemin et/ou le nom du fichier.

- **Variable.** Définit la variable qui contient le flux d'impression. Le contenu de la variable sélectionnée est utilisé.
- **Personnalisé.** Définit le contenu personnalisé. Vous pouvez utiliser le contenu fixe, un mixte de contenu fixe et variable, ou contenu variable seul. Pour insérer un contenu de variable, cliquez le bouton avec une flèche à droite de la zone de données et insérez la variable de la liste. Pour plus d'informations, consultez l'article Utiliser des valeurs composées.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  > **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Envoyer Les Données Au Port Série

Envoie les données vers un port série Vous pouvez utiliser cette action pour la communication avec les appareils externes à port série. Assurez-vous que la configuration du port série est identique des deux côtés, dans l'action et dans l'appareil à port série. Le port série peut être utilisé par une application dans la machine. Pour une utilisation du port avec succès pour cette action, aucune autre application ne peut utiliser le port, même pas un pilote d'imprimante.

Port.

- **Nom de Port.** Spécifie le nom du port sur lequel se connecte votre appareil externe. Cela peut être un port COM matériel ou un port COM virtuel.

Paramètres de Port

Cette section affiche les options pour la connexion du port série. Assurez-vous que les paramètres correspondent aux paramètres de votre appareil extérieur.

- **Bits par seconde.** Spécifie la vitesse qui sera utilisée par l'appareil extérieur pour communiquer avec l'ordinateur. L'alias généralement utilisé pour ce paramètre est "baud rate".

- **Bits de données.** Spécifie le nombre de bits de données dans chaque caractère. 8 bits de données sont généralement utilisés dans les appareils récents.
- **Parité.** Spécifie la méthode de détection d'erreurs de transmission. Le paramètre de parité généralement utilisé est "aucune", avec la détection d'erreur gérée par un protocole de communication (contrôle de flux).
- **Bits d'arrêt.** Les bits d'arrêt envoyés à la fin de chaque caractère permettent à la machine de réception de détecter la fin d'un caractère et de le resynchroniser avec le flux de caractères. Les appareils électroniques utilisent généralement un bit de stop.
- **Contrôle de flux.** Un port série peut utiliser des signaux dans l'interface pour suspendre et relancer la transmission de données.

  > **EXEMPLE:** Par exemple, un appareil lent peut avoir besoin d'utiliser la poignée de main avec le port série pour indiquer que les données auraient du être suspendues pendant que les processus de l'appareil reçoivent des données.

Contenus

Cette section vous permet de définir le contenu que vous voulez envoyer au port série. Vous pouvez utiliser le contenu fixe, un mixte de contenu fixe et variable, ou contenu variable seul. Pour insérer un contenu de variable, cliquez le bouton avec une flèche à droite de la zone de données et insérez la variable de la liste. Pour plus d'informations, consultez l'article Utiliser des valeurs composées.

- **Données.** Spécifie le contenu qui sera envoyé en sortie.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  > **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Envoyer Les Données Au Port TCP/IP

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Envoie les données à tout appareil externe acceptant une connexion TCP/IP sur un numéro de port prédéfini. Cette action établit la connexion avec l'appareil, envoie les données et termine la connexion. La connexion et la communication sont gérées par le protocole de communication qui s'établit entre le client et le serveur au début et à la fin de la connexion TCP.

Paramètres de Connexion

NOTE: Cette action supporte le Protocole Internet version 6 (IPv6).

- **Destination.** Définit l'adresse et le port de destination du serveur TCP/IP. Vous pouvez coder-en-dur les paramètres de connexion et utiliser un nom d'hôte et une adresse IP fixes. Vous pouvez également utiliser des paramètres de connexion variables. Pour plus d'informations, consultez l'article Utiliser des valeurs composées.

EXEMPLE: Si la variable `hostname` procure le nom du serveur TCP/IP et la variable `port` fournit le numéro de port, vous pouvez entrer la destination suivante :
`[hostname]:[port] (nom d'hôte)`

- **Délai de déconnexion.** Prolonge la connexion sur le socket ciblé de l'intervalle de temps défini après que les données ont été fournies. Certains appareils ont besoin de plus de temps pour traiter les données. Par défaut, le délai est désactivé.
- **Répondre à l'expéditeur.** Active la réponse directe au socket d'où les données de déclencheur proviennent. Cette option peut être utilisée pour envoyer un retour du processus d'impression.

  Conditions préalables pour les paramètres de réponse à l'expéditeur

  Les conditions préalables suivantes sont requises :

  - Le correspondant distant ne déconnecte pas le canal de communication dès que le message a été délivré.
  - L'action **Envoyer les Données au port TCP/IP** est utilisée dans le déclencheur du **Serveur TCP/IP**.
  - L'**Évènement d'Exécution** dans le déclencheur du serveur TCP/IP n'est pas configuré en **A la déconnexion du client**.

Contenus

Cette section vous permet de définir le contenu que vous voulez envoyer au serveur TCP/IP. Vous pouvez utiliser le contenu fixe, un mixte de contenu fixe et variable, ou contenu variable seul. Pour insérer un contenu de variable, cliquez le bouton avec une flèche à droite de la zone de données et insérez la variable de la liste. Pour plus d'informations, consultez l'article Utiliser des valeurs composées.

- **Données.** Spécifie le contenu qui sera envoyé en sortie.
- **Encoding. (encodage)** Spécifie l'encodage des données envoyées.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  > **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Définir Le Nom De La Tâche D'impression

Spécifie le nom de la tâche d'impression comme il apparaît dans le spouleur Windows. Par défaut, le nom de la tâche d'impression est le nom du fichier d'étiquette utilisé et cette action va le substituer. Vos devez toujours imbriquer l'action sous l'action **Open Label**, pour l'associer au fichier d'étiquette spécifique.

Tâches d'Impression

- **Nom.** Spécifie le nom de la tâche. Il peut être codé-en-dur, et le même nom sera utilisé pour chaque action d'Impression d'Étiquette. L'option **Variable** active le nom de fichier variable. Vous devez sélectionner une variable qui contient le chemin et/ou le nom du fichier quand le déclencheur est exécuté. En général, la valeur vers la variable est assignée par un filtre.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

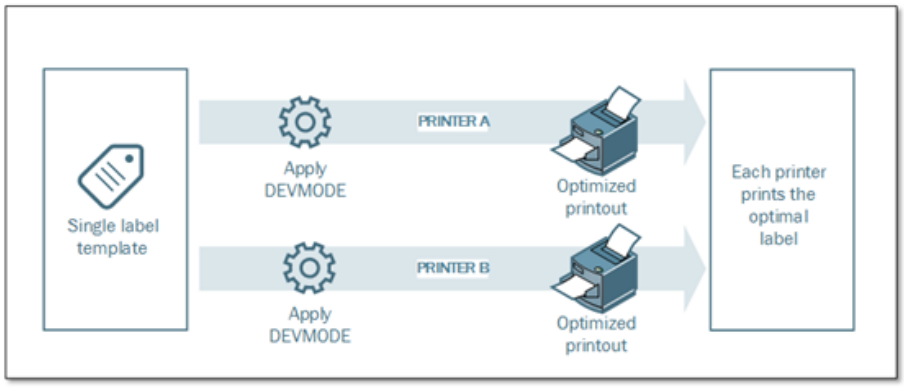
EXEMPLE: A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Définir Les Paramètres D'Impression

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Permet d'affiner les paramètres relatifs au pilote d'imprimante, tels que la vitesse et l'obscurité des imprimantes d'étiquettes, ou le réservoir à papier pour les imprimantes laser. Les paramètres d'imprimante sont appliqués uniquement pour l'impression actuelle et ne sont pas mémorisés pour l'évènement de déclenchement suivant.



Si vous utilisez l'action Définir l'imprimante (définir l'imprimante) pour changer le nom d'imprimante, assurez-vous d'utiliser l'action **Set Print Parameter**(définir le paramètre d'impression) ensuite. Avant de pouvoir appliquer la structure DEVMODE au pilote d'imprimante, vous devez commencer par charger les paramètres d'imprimante par défaut, ce qui sera effectué par l'action Set Printer. Le DEVMODE est seulement compatible avec le DEVMODE du même pilote d'imprimante.

Paramètres d'Impression

Cette section définit les paramètres disponibles pouvant être affinés avant l'impression.

- **Réservoir à Papier.** Définit le nom du réservoir à papier contenant le média d'étiquette. Utilisé généralement avec les imprimantes laser et à jet d'encre ayant différents réservoirs à papier. Le nom du réservoir à papier fourni doit correspondre au nom du réservoir dans le pilote d'imprimante. Pour plus d'informations, voir caractéristiques du pilote d'imprimante.
- **Vitesse d'impression.** Définit la valeur pour la vitesse d'impression et substitue les paramètres de l'étiquette. La valeur fournie doit se situer dans la gamme de valeurs acceptables. Par exemple, un modèle d'imprimante accepte une gamme de valeurs de 0 à 30, l'autre modèle

d'imprimante accepte des valeurs de -15 à 15. Pour plus d'informations, voir caractéristiques du pilote d'imprimante.

- **Obscurité.** Définit l'obscurité des objets imprimés sur le papier et substitue les paramètres de l'étiquette. La valeur fournie doit se situer dans la gamme de valeurs acceptables. Pour plus d'informations, voir caractéristiques du pilote d'imprimante.
- **Décalage d'impression X.** Applique le décalage horizontal. L'impression de l'étiquette sera repositionnée pour le nombre de points spécifiés en direction horizontal. Vous pouvez définir un décalage négatif.
- **Décalage d'impression Y.** Applique le décalage vertical. L'impression de l'étiquette sera repositionnée pour le nombre de points spécifiés en direction verticale. Vous pouvez définir un décalage négatif.

L'option **Variable** à côté de chaque paramètre active le contenu de la variable. Vous devez sélectionner une variable qui contient la valeur du paramètre sélectionné quand le déclencheur est exécuté.

Paramètres Avancés d'Impression

Cette section personnalise les paramètres d'impression envoyés avec la tâche d'impression. Tous les paramètres d'impression, tels que la vitesse d'impression, l'obscurité, le type de média, le décalage et similaires, sont généralement définis soit (1) dans l'étiquette propre, ils sont (2) requis par le pilote d'imprimante, ou (3) requis par l'imprimante à l'impression. Les méthodes supportées dépendent du pilote d'imprimante et des possibilités de l'imprimante.

Le mode d'impression (requête de paramètres de l'étiquette, du pilote ou de l'imprimante) est configurable dans la conception de l'étiquette (voir l'onglet Fichier>Caractéristiques d'étiquette>Imprimante).

Vous pouvez toutefois être dans le besoin d'appliquer ces paramètres d'imprimante au moment de l'impression, et ils peuvent être différents pour chaque impression.

> **EXEMPLE:** Par exemple, vous voulez imprimer un modèle d'étiquette unique (fichier .LBL) à différentes imprimantes mais chaque imprimante nécessite des paramètres légèrement différents. Les imprimantes de différents fabricants n'utilisent pas les mêmes valeurs de vitesse ou de température. De plus, certaines imprimantes requièrent un décalage vertical ou horizontal pour imprimer l'étiquette au bon endroit. Durant la phase de test, vous pouvez déterminer les meilleurs paramètres pour chaque imprimante que vous allez utiliser et les appliquer à un modèle d'étiquette unique juste avant d'imprimer. Cette action va appliquer les paramètres correspondants pour chaque imprimante définie.

L'action requiert la réception des paramètres d'impression sous une structure DEVMODE. C'est une structure de données standard Windows contenant les informations d'initialisation et d'environnement d'une imprimante. Pour plus d'informations, voir l'article Comprendre les paramètres d'imprimante et DEVMODE.

L'option **Paramètres d'Imprimante** appliquera les paramètres d'imprimante personnalisés. Vous pouvez utiliser les entrées suivantes :

1. **Fixed-data Base64-encoded DEVMODE. (donnée-fixe DEVMODE encodé-Base64)** Dans ce cas, vous devez fournir le DEVMODE de l'imprimante encodé dans une chaîne de caractères Base64 directement dans le champ d'édition. Quand l'action est exécutée, elle convertira les données encodées en Base64 sous forme binaire.

2. **Variable-data Base64-encoded DEVMODE. (donnée-variable DEVMODE encodé-Base64)** Dans ce cas la variable sélectionnée doit contenir le DEVMODE encodé Base64. Activer la **Variable** et sélectionnez la variable appropriée dans la liste. Quand l'action est exécutée, elle convertira les données encodées en Base64 sous forme binaire.
3. **Variable-data binary DEVMODE. (donnée-variable DEVMODE binaire)** Dans ce cas, la variable sélectionnée doit contenir le DEVMODE sous sa forme binaire native. Activer la **Variable** et sélectionnez la variable appropriée dans la liste. Quand elle est exécutée, l'action va utiliser le DEVMODE tel quel, sans aucune conversion.

NOTE: Si la variable fournit un DEVMODE binaire, assurez-vous que la variable sélectionnée est définie comme **variable binaire** dans la configuration du déclencheur.

Extraction de la structure DEVMODE

La structure DEVMODE est encodée dans le registres du système. Vous pouvez l'extraire des registres.

Utilisez l'action **Set Printer Parameter** pour vous aider à tester, l'application qui a été fournie va récupérer le DEVMODE de l'imprimante sélectionnée et va vous le sauvegarder dans un fichier ou l'encoder en Base64. Vous trouverez l'application `GetPrinterSettings.exe` sur le site Web NiceLabel Automation DVD et en-ligne NiceLabel.

Utilisation interactive de l'application

Lancez l'application, sélectionnez l'imprimante pour laquelle vous avez besoin de la structure DEVMODE et cliquez le bouton **Obtenir les Paramètres d'Imprimante**. Le DEVMODE sera fourni sous une chaîne de caractères Base64. Vous pouvez le coller dans l'action **Définir les Paramètres d'Impression**.

Utiliser l'application avec les paramètres de la ligne de commande

Dans ce cas, vous pouvez contrôler l'application avec les paramètres de la ligne de commande.

Syntaxe :

```
GetPrinterSettings.exe <printer_name> <file_name> [base64]
```

- `printer_name`: le nom du pilote d'imprimante est disponible dans le système Windows.
- `file_name`: nom du fichier qui contiendra le DEVMODE extrait
- `base64`: paramètre optionnel. S'il est fourni, le DEVMODE sera encodé dans la chaîne de caractères Base64, autrement le DEVMODE sera fourni comme donnée binaire

Par exemple :

Sauvegarder le DEVMODE pour l'imprimante "Avery AP 5.4 300DPI" comme données binaires dans le fichier "devmode1".

```
EXEMPLE: GetPrinterSettings.exe "Avery AP 5.4 300DPI" c:\temp\devmode1
```

Sauvegarder le DEVMODE pour l'imprimante "Avery AP 5.4 300DPI" comme données encodées Base64 dans le fichier "devmode2".

```
EXEMPLE: GetPrinterSettings.exe "Avery AP 5.4 300DPI" c:\temp\devmode2 base64
```

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  > **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Définir L'imprimante

Spécifie le nom de l'imprimante sur laquelle l'étiquette sera imprimée. Utilisez cette action pour substituer l'imprimante définie dans le modèle d'étiquette. Cette action est utile quand vous devez imprimer le même modèle d'étiquette sur différentes imprimantes. Vous devez toujours imbriquer cette action sous l'action Open Label - Ouvrir l'étiquette (ouvrir l'étiquette) pour référencer l'étiquette pour laquelle l'imprimante est changée. Cette action lit les paramètres par défaut --tels que la vitesse et l'obscurité-- du pilote d'imprimante sélectionné et les applique à l'étiquette. Si vous n'utilisez pas l'action Set Printer, l'étiquette sera imprimée comme définie dans le modèle d'étiquette.

> **AVERTISSEMENT** Soyez prudent quand vous changez d'une marque d'imprimante à l'autre, par ex. de Zebra à SATO, ou même d'un modèle d'imprimante à un autre modèle de la même marque. Les paramètres d'imprimante ne doivent pas être compatibles et l'impression de l'étiquette ne doit pas être identique. De même, les optimisations de conception de l'étiquette pour l'imprimante originale, tels que les compteurs internes et types de caractères internes ne doivent pas être disponibles pour l'imprimante sélectionnée.

Imprimante

- **Nom d'imprimante.** Spécifie le nom d'imprimante. Vous pouvez sélectionner l'imprimante dans la liste d'imprimantes installées localement, ou vous pouvez entrer un nom d'imprimante. L'option **Variable** active l'imprimante variable. Quand activée, vous devez sélectionner une variable qui contiendra le nom d'imprimante quand le déclencheur est exécuté. En général, la valeur vers la variable est assignée par un filtre.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.

- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  EXEMPLE: A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Définir Les Variables

Assigne une nouvelle valeur à la variable sélectionnée. Généralement, les variables vont recevoir leur valeur par l'action Utilisation d'un Filtre de Données, qui va extraire les champs des données reçues et les mapper sur les variables. Vous aurez peut être besoin de définir les valeurs de variables vous-même, d'habitude à des feins de résolution de problèmes. Les valeurs de variables ne sont pas mémorisées d'un déclencheur à l'autre, mais sont conservées durant l'utilisation du déclencheur.

Variable

Cette section vous permet de définir le contenu que vous voulez assigner à la variable sélectionnée.

- **Nom.** Spécifie le nom de la variable qui recevra une nouvelle valeur.
- **Valeur.** Spécifie la nouvelle valeur de la variable. Vous pouvez utiliser le contenu fixe, un mixte de contenu fixe et variable, ou contenu variable seul. Pour insérer un contenu de variable, cliquez le bouton avec une flèche à droite de la zone de données et insérez la variable de la liste. Pour plus d'informations, consultez l'article Utiliser des valeurs composées.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

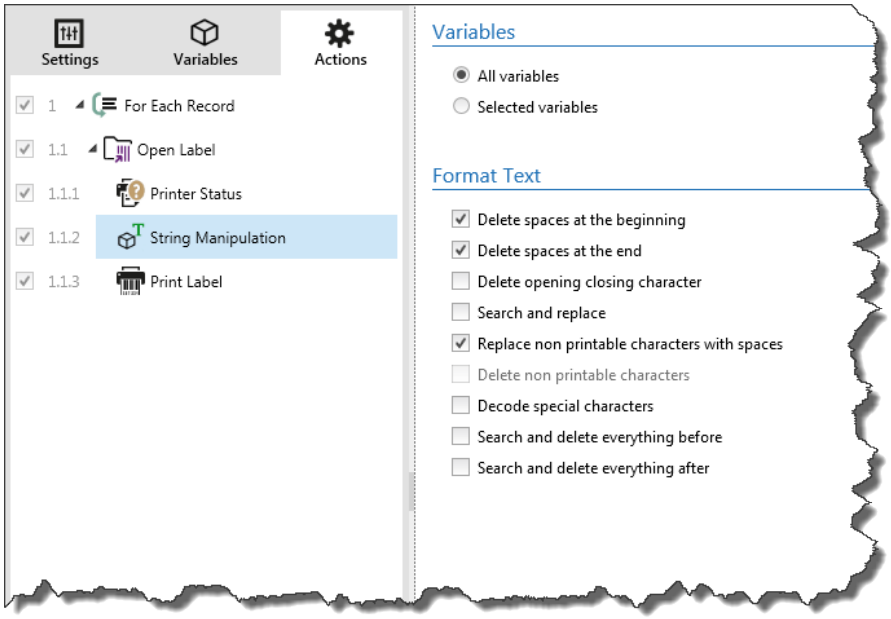
> **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Manipulation De Chaîne De Caractères

Formate les valeurs des variables sélectionnées. Vous pouvez réaliser les actions telles que : effacer les espaces de début et de fin, rechercher et remplacer des caractères, effacer les parenthèses d'ouverture et de fermeture. Généralement vous aurez besoin de cette fonctionnalité quand le déclencheur reçoit des données non-structurées ou brutes, et vous avez besoin de les analyser avec le filtre **Données non-structurées**. Cette action vous permet d'affiner la valeur des données.

> **NOTE:** Si cette action ne vous procure pas suffisamment de pouvoir de manipulation d'une chaîne de caractères particulière, vous pouvez utiliser l'action **Exécuter le Script** et utiliser Visual Basic Script ou Python pour manipuler vos données.



Variables

Cette section définit les variables pour lesquelles les manipulations de chaîne de caractères s'appliqueront.

- **All variables. (toutes les variables)** Spécifie que la(les) manipulation(s) s'appliquera à toutes les variables définies.
- **Selected variables. (variables sélectionnées)** Spécifie que la(les) manipulation(s) s'appliquera à toutes les variables sélectionnées.

Options de Formatage

Cette section définit les fonctions de manipulation de chaîne de caractères qui seront appliquées aux

variables ou champs sélectionnés. Vous pouvez sélectionner une ou plusieurs fonctions. Les fonctions seront appliquées dans l'ordre sélectionné dans l'interface utilisateur, de haut en bas.

- **Enlever les espaces au début.** Enlève tous les caractères d'espacement (code décimal ASCII 32) du début de la chaîne de caractères.
- **Enlever les espaces à la fin.** Enlève tous les caractères d'espacement (code décimal ASCII 32) à la fin de la chaîne de caractères.
- **Enlever les caractères d'ouverture et fermeture.** Efface la première occurrence du caractère d'ouverture et de fermeture trouvé dans la chaîne de caractères.

  EXEMPLE: Si vous utilisez "{" comme caractère d'ouverture et "}" comme caractère de fermeture, la chaîne d'entrée `{{selection}}` sera convertie en `{selection}`.

- **Rechercher et remplacer.** Exécute une recherche classique et remplace la fonction selon la valeur fournie pour *find what* et *replace with* (chercher-quoi et remplacer-par). Vous pouvez utiliser des expressions classiques.

  NOTE: Il y a plusieurs implémentations des expressions classiques utilisées. NiceLabel Automation utilise la syntaxe .NET Framework pour les expressions classiques. Pour plus d'informations, consultez la Base de Connaissances article KB250.

- **Remplacer les caractères non-imprimables par un espace.** Remplace tous les caractères de contrôle de la chaîne de caractères par des caractères d'espacement (code décimal ASCII 32). Les caractères non-imprimables sont des caractères à valeurs ASCII entre 0-31 et 127-159.
- **Effacer les caractères non-imprimables.** Efface tous les caractères dans la chaîne de caractères. Les caractères non-imprimables sont des caractères à valeurs ASCII entre 0-31 et 127-159.
- **Décoder les caractères spéciaux.** Les caractères spéciaux (ou caractères de contrôle) sont des caractères qui ne sont pas disponibles sur le clavier, tels que Retour Charriot et Passage à la Ligne. NiceLabel Automation utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Charriot et <LF> pour Passage à la Ligne. Pour plus d'informations, consultez l'article Introduction de Caractères Spéciaux (Codes de Contrôle). Cette option convertit les caractères spéciaux de la syntaxe NiceLabel en caractères binaires réels.

  EXEMPLE: Quand vous recevez les données "<CR><LF>", NiceLabel Automation les utilisera comme une chaîne complète de 8 caractères. Pour interpréter et utiliser les données comme deux caractères binaires `CR` (Retour Charriot - code ASCII 13) et `LF` (Passage à la Ligne - code ASCII 10), vous devrez activer cette nouvelle option.

- **Rechercher et effacer tout avant.** Trouve la chaîne de caractères fournie et efface tous les caractères du début des données jusqu'à la chaîne de caractères. La chaîne de caractères trouvée peut aussi être effacée.
- **Rechercher et effacer tout après.** Trouve la chaîne de caractères fournie et efface tous les caractères depuis la chaîne de caractères jusqu'à la fin des données. La chaîne de caractères trouvée peut aussi être effacée.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront

exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.

- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

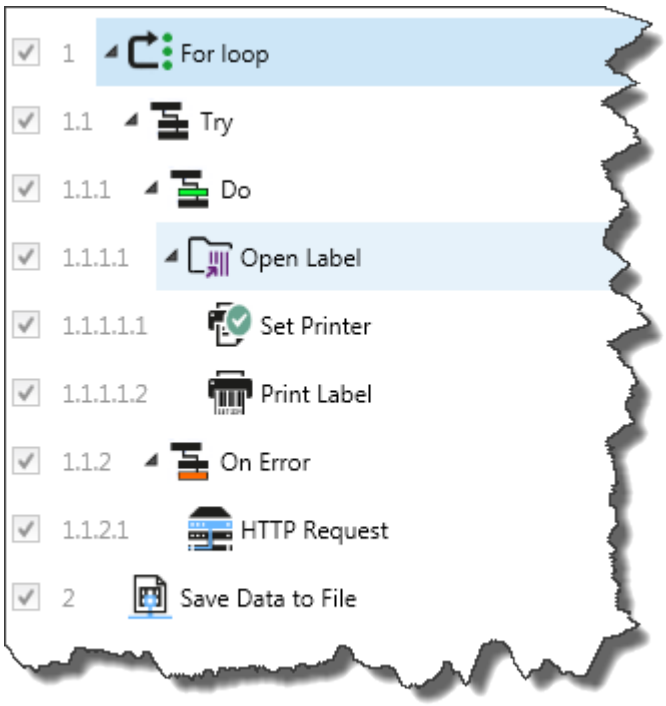
  > **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Essai

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Enterprise**.

Permet une détection facile des erreurs durant l'exécution, et l'exécution d'un autre ensemble d'actions, si l'erreur apparaît. L'action crée des espaces réservés **Do (faire)** et **On error** (pour erreur) pour des actions. Toutes les actions qui doivent s'effectuer quand le déclencheur est activé doivent être placées dans l'espace réservé Do. Si aucune erreur n'est détectée durant l'exécution des actions de l'espace réservé Do, elles seront les seules actions à être exécutées. Toutefois, quand une erreur survient, l'exécution des actions de l'espace réservé Do vont s'interrompre et l'exécution passe aux actions de l'espace réservé On error.



EXEMPLE: Si une des actions dans l'espace réservé Do échoue, l'exécution de l'action s'arrête et redémarre avec les actions dans l'espace réservé Pour Erreur. Si Try (essai) est placé séparément, il terminera l'exécution du déclencheur. Dans ce cas, Try est imbriqué sous l'action For loop (en boucle). Normalement, toute erreur dans l'espace réservé Do arrêtera aussi l'exécution de l'action For loop (en boucle), même si d'autres phases de For loop doivent encore être exécutées. Dans ce cas, Sauver les Données dans un Fichier ne sera pas exécutée non plus. Par défaut, toute erreur interromp le processus complet du déclencheur.

Vous pouvez toutefois continuer l'exécution de l'itération suivante de l'action For loop. Pour cela, vous devez activer Ignorer les défauts dans l'action Try. Si les données de la phase actuelle de For Loop causent une erreur dans l'emplacement réservé, les action de Pour Erreur seront exécutées, ensuite la Sauvegarde des Données dans un Fichier de niveau 2 sera exécutée et ensuite l'action For loop continuera l'exécution de l'itération suivante.

Cette action procure une détection d'erreur facile et l'exécution d'actions "feedback" ou "reporting". Par exemple, si une erreur survient durant le traitement du déclencheur, vous pouvez envoyer un avertissement. Pour plus d'informations, consultez l'article Feedback de la tâche d'impression.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action

actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

> **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

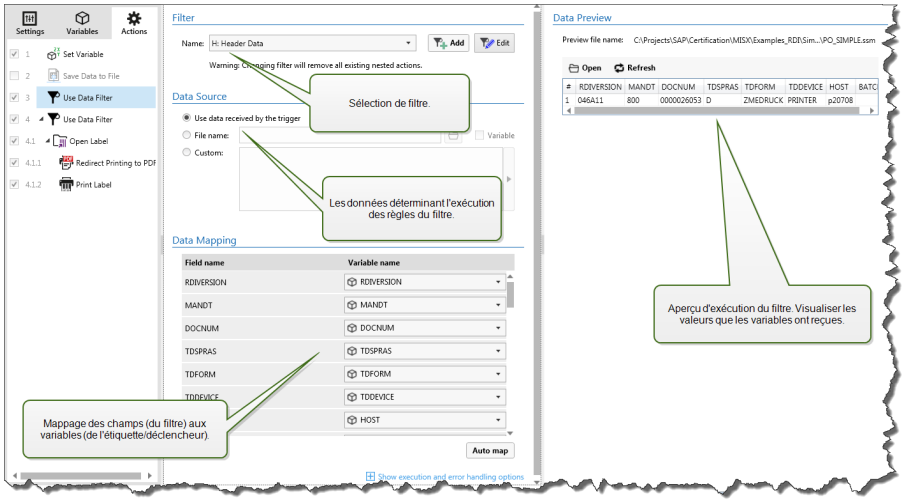
Utilisation D'un Filtre De Données

Exécute les règles du filtre sur la source de données d'entrée. En résultat, l'action va extraire les champs des données d'entrée et mapper leurs valeurs dans les variables associées. Donc, l'action exécute le filtre sélectionné et assigne les valeurs respectives aux variables.

- **Éléments de niveau inférieur.** L'action peut créer des éléments de sous-niveau, identifiés par **"pour chaque ligne"** ou **"pour chaque bloc de données..."**. Quand vous les voyez, le filtre va extraire les données, pas au niveau du document (avec des placement de champs codés-en-dur), mais de façon relative depuis les sous ensembles qui contiennent des sections répétables. Dans ce cas, assurez-vous de positionner vos actions sous ces éléments. Vous devez imbriquer l'action sous tels éléments.
- **Mapper les variables aux champs.** Le mappage entre les variables du déclencheur et les champs du filtre est défini soit manuellement, ou est automatisé, selon la configuration du filtre. Si vous avez défini les champs du filtre manuellement, vous devez également mapper les champs aux variables correspondantes manuellement.

> Il est conseillé de définir les champs en utilisant les mêmes noms que les variables de l'étiquette. Dans ce cas, le bouton **Mappage Auto** va mapper les noms correspondants automatiquement.

- **Tester l'exécution du filtre.** Quand le mappage des champs de variables est terminé, vous pouvez tester l'exécution du filtre. Le résultat sera affiché à l'écran dans une table. Le nombre de lignes dans la table représente le nombre de fois que les actions seront exécutées dans le niveau sélectionné. Les noms de colonnes représentent les noms de variables. La cellule contient les valeurs comme assignées à la variable respective par le filtre. Le nom de fichier d'aperçu par défaut est hérité de la définition du filtre, vous pouvez exécuter le filtre sur tout les fichiers.



Pour plus de renseignements, consultez l'article Comprendre les Filtres et l'article Exemples.

Filtre

- **Nom.** Spécifie le nom du filtre que vous voulez appliquer. La liste contient tous les filtres définis dans la configuration actuelle. Vous pouvez utiliser les trois éléments du bas de la liste pour créer un nouveau filtre.

NOTE: La sélection d'un autre filtre va enlever toutes les actions imbriquées sous cette action. Si vous voulez conserver les action actuellement définies, déplacez-les en dehors de l'action **Use Data Filter**. Si vous avez perdu vos actions, vous pouvez annuler votre action et revenir à la configuration précédente.

Source de Données

Cette section vous permet de définir le contenu que vous voulez envoyer à l'imprimante.

- **Utiliser les données reçues par le déclencheur.** Définit que les données reçues par le déclencheur sont utilisées dans un filtre. Dans ce cas, l'action va utiliser les données originales reçues par le déclencheur et exécuter les règles du filtre sur celle-ci.

Par exemple, si vous utilisez un déclencheur à fichier, les données sont le contenu du fichier surveillé. Si vous utilisez un déclencheur de base de données, les données seront un ensemble de données fournies par la base de données. Si vous utilisez un déclencheur TCP/IP, les données sont le contenu brut reçu sur un socket.

- **File name. (nom de fichier)** Définit le chemin et le nom du fichier contenant les données selon lesquelles les règles du filtre seront exécutées. Le contenu du fichier spécifié est utilisé dans un filtre. L'option **Variable** active le nom de fichier variable. Vous devez sélectionner une variable qui contient le chemin et/ou le nom du fichier.
- **Personnalisé.** Définit le contenu personnalisé à analyser par le filtre. Vous pouvez utiliser le contenu fixe, un mixte de contenu fixe et variable, ou contenu variable seul. Pour insérer le contenu d'une variable, cliquez le bouton avec une flèche à droite de la zone de données et insérez la variable de la liste. Pour plus d'informations, consultez l'article Utiliser des valeurs composées.

Aperçu des données

Cette section fournit un aperçu de l'exécution du filtre. Le contenu d'aperçu du nom de fichier est lu

et le filtre sélectionné y est appliqué. Les règles du filtre vont extraire les champs. La table va afficher le résultat de l'extraction. Chaque ligne de la table représente les données pour une étiquette. Chaque colonne représente la variable. Avant de voir les résultats, vous devez configurer le mappage des champs au variables respectives. En fonction de la définition du filtre, vous pouvez mapper les variables aux champs manuellement, ou c'est fait automatiquement.

- **Aperçu nom de fichier.** Spécifie le fichier qui contient l'échantillon de données qui sera analysé dans le filtre. Le fichier d'aperçu est copié de la définition du filtre. Si vous changez le nom du fichier d'aperçu, le nouveau nom de fichier sera sauvegardé.
- **Ouvrir.** Sélectionne un autre fichier selon lequel vous voulez exécuter les règles du filtre.
- **Rafraîchir.** Relance les règles du filtre en fonction du contenu du nom du fichier d'aperçu. La section Aperçu de Données sera mise à jour avec le résultat.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  > **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Vérifier La Licence

Lit la licence activée et exécute les actions imbriquées sous cette action seulement si un certain type de licence est utilisé.

Cette action assure une protection pour que la configuration de vos déclencheurs ne puisse pas être utilisée sur des machines non-autorisées. La clé de licence qui active le logiciel peut également encoder une Solution ID. C'est un nombre unique qui identifie le fournisseur de solution qui a vendu la licence NiceLabel Automation. Si la Solution ID configurée correspond à la Solution ID encodée dans la licence, la machine de destination sera autorisée d'utiliser les actions imbriquées, limitant effectivement l'exécution aux licences vendues par le fournisseur de la solution.

Les déclencheurs peuvent être cryptés et verrouillés pour que seuls les utilisateurs autorisé puisent ouvrir la configuration. Pour plus d'informations, voir l'article Protéger la Configuration du Déclencheur contre l'édition.

Information de licence

- **Solution ID.** Définit le numéro ID des licences qui sont autorisées à effectuer les actions imbriquées.
    - Si la valeur introduite n'est pas la Solution ID lue de la licence, l'action imbriquée ne sera pas exécutée.
    - Si la valeur introduite est 0, les actions seront seulement exécutées si une licence valable est trouvée.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

    > **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Service Web

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Enterprise**.

Se connecte à un serveur SOAP et exécute les méthodes sur celui-ci. Cette action va envoyer des données sortantes au service Web et recueillir les résultats.

Vous devez imprimer les étiquettes de produit. Votre déclencheur ne recevra que le segment de données nécessaire. Par ex. le déclencheur reçoit la valeur pour `ID Produit` et `Description`, mais pas le `Prix`. L'information de prix est disponible dans une base de données séparée qui est accessible par un appel du service Web. Le service Web définit la fonction par sa définition WSDL, telle que l'entrée de la fonction est `ID Produit` et la sortie est `Prix`. L'action du service Service

va envoyer `ID Produit` au service Web, qui va effectuer une recherche interne de sa base de données et fournir le `Prix` correspondant comme résultat. L'action sauvegardera le résultat dans la variable, qui peut être utilisée sur l'étiquette.

Définition du service Web

NOTE: Cette action supporte le Protocole Internet version 6 (IPv6).

- **WSDL.** Spécifie l'emplacement de la définition Web Service Description Language (WSDL). C'est une description du langage d'interface sur base XML qui décrit les fonctionnalités offertes par le service Web. Le WSDL est généralement fourni par le service Web lui-même. Typiquement, vous introduisez le lien vers le WSDL et cliquez le bouton **Importer** pour lire la définition. Si vous avez du mal a obtenir la WSDL de la ressource en-ligne, sauvegardez le WSDL dans un fichier et entrer le chemin et le nom du fichier pour en charger les méthodes.
  NiceLabel Automation détectera automatiquement si le service Web distant utilise la syntaxe **document** ou **RPC** et communique de façon appropriée.
- **Adresse.** Procure l'adresse où le service Web est publié. Initialement, cette information est extraite de WSDL, mais vous pouvez la mettre à jour avant l'exécution de l'action. C'est utile pour les environnements de développement / test / production, où vous utilisez la même liste d'actions, mais avec des noms différents de serveurs où le service Web fonctionne.

  Vous pouvez utiliser le contenu fixe, un mixte de contenu fixe et variable, ou contenu variable seul. Pour insérer un contenu de variable, cliquez le bouton avec une flèche à droite de la zone de données et insérez la variable de la liste. Pour plus d'informations, consultez l'article Utiliser des valeurs composées.
- **Méthode.** Liste les méthodes (fonctions) disponibles sur le service Web sélectionné. La liste est mise à jour automatiquement par la définition WSDL.
- **Paramètres.** Définit les variables d'entrée et de sortie pour la méthode (fonction) sélectionnée. Les paramètres d'entrée attendent une entrée du déclencheur. Pour des raisons de détection d'erreurs et de test vous pouvez introduire des valeurs fixes et visualiser le résultat à l'écran. Mais typiquement vous allez sélectionner une variable pour le paramètre entrant. La valeur de cette variable sera utilisée comme paramètre d'entrée. Le paramètre de sortie fournit le résultat de la fonction. Vous devez sélectionner la variable qui va sauvegarder le résultat.
- **Timeout. (délai d'expiration)** Définit le délai durant lequel le serveur essaiera d'établir la connexion.

Authentification d'Utilisateur

- **Activer l'authentification de base.** Définit les infos d'identification nécessaires à établir l'appel externe vers le service Web distant. Pour plus d'informations concernant la sécurité, consultez l'article Sécuriser l'accès à vos déclencheurs.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est

une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.

- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

  EXEMPLE: A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Aperçu des données

- **Exécuter.** Exécute l'appel du service Web. Il envoie les valeurs des paramètres de sortie au service Web et fournit le résultat dans le paramètre de sortie. Utilisez cette fonctionnalité pour tester l'exécution du service Web. Vous pouvez introduire des valeurs pour les paramètres de sortie et visualiser le résultat à l'écran. Quand vous êtes satisfait de l'exécution, vous remplacerez les values fixes introduites du paramètre d'entrée par une variable de la liste.

Transformation XML

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Enterprise**.

Transforme le document XML en un autre document en utilisant les règles de transformation fournies. Les règles doivent être fournies par la définition .XSLT dans un fichier, ou par une une autre source variable. L'action permet de convertir les documents XML complexes en documents XML ayant une structure plus gérable. XSLT signifie Transformations XSL. XSL signifie EXtensible Stylesheet Language, ce qui est un langage en style de page pour les documents XML.

L'action va sauvegarder le document XML convertit dans la variable sélectionnée. Le fichier original reste intact sur le disque. Si vous voulez sauvegarder le document XML convertit, utilisez l'action Sauvegarder les données dans un fichier.

Typiquement, vous utiliserez l'action pour simplifier les documents XML fournis par l'application hôte. Définir des filtres XML pour les document XML complexes peut prendre un certain temps, dans certains cas le XML est trop complexe pour être traité. Comme alternative, vous pouvez définir les règles pour convertir XML en une structure qui peut être traitée facilement par le filtre XML, ou même éliminer complètement la nécessité d'un filtre. Vous pouvez convertir un document XML en un XML supporté de façon native, tel que XML Oracle et ensuite l'exécuter simplement avec l'action Utiliser un fichier de Commande Oracle XML.

Source de Données

Cette section définit les données XML que vous voulez transformer.

- **Utiliser les données reçues par le déclencheur.** Définit que les données reçues par le déclencheur seront utilisées. Le même résultat peut être atteint en activant la variable interne `DataFileName` et en utilisant le contenu du fichier auquel elle se réfère. Pour plus d'informations, consultez l'article Variables Internes.
- **Nom de fichier.** Définit le chemin et nom du dossier contenant le fichier XML à transformer. Le contenu du fichier spécifié est en usage. L'option **Variable** active le nom de fichier variable. Vous devez sélectionner une variable qui contient le chemin et/ou le nom du fichier. Cette action va ouvrir le fichier spécifié et appliquer la transformation sur le contenu du fichier, qui doit être en format XML.
- **Variable.** Définit la variable qui contient le flux d'impression. Le contenu de la variable sélectionnée est utilisé et doit contenir une structure XML.

Source de Données des Règles de Transformation (XSLT)

Cette section définit les règles de transformation (document .XSLT) qui seront appliquées au document XSLT.

- **Nom de fichier.** Définit le chemin et nom du dossier contenant les règles de transformation (.XLST).
- **Personnalisé.** Définit le contenu personnalisé. Vous pouvez utiliser le contenu fixe, un mixte de contenu fixe et variable, ou contenu variable seul. Pour insérer un contenu de variable, cliquez le bouton avec une flèche à droite de la zone de données et insérez la variable de la liste. Pour plus d'informations, consultez l'article Utiliser des valeurs composées.

Sauver le Résultat dans une Variable

- **Variable.** Spécifie la variable qui contiendra le résultat du processus de transformation. Par ex. si vous utilisez les règles qui convertissent un XML complexe en un XML plus simple, le contenu de la variable sélectionnée est l'XML simple.

Exécution d'Action et Traitement d'Erreur

- **Activé.** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Vous pouvez utiliser cette fonctionnalité pour les essais.
- **Condition.** Définit une expression en-ligne de programmation qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action sera exécutée. C'est une méthode pour ne pas exécuter les actions chaque fois, mais seulement quand les variables surveillées ont une certaine valeur.
- **Ignorer le défaut.** Spécifie d'ignorer le défaut et continuer par l'action suivant, même si l'exécution de l'action actuelle échoue. Les actions imbriquées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution de l'action continuera avec l'action suivante au même niveau que l'action actuelle. L'erreur est sauvegardée, mais elle n'interrompt pas l'exécution de l'action. Pour plus d'informations, consultez l'article Traitement d'Erreurs

> **EXEMPLE:** A la fin de l'impression vous pouvez vouloir envoyer la mise à jour du résultat à une application externe qui utilise l'action Requête HTTP. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapportage, même après un échec d'impression, l'action Imprimer l'Étiquette doit avoir l'option "Ignorer les Échecs" activée.

- **Sauvegarder l'erreur dans une variable.** Spécifie la sauvegarde de la description d'erreur dans une variable, quand une erreur interrompt l'exécution de cette action. La même cause d'erreur est aussi sauvegardée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`. Pour plus d'informations, consultez l'article Variables Internes.

Exemple

L'exemple de cette action est installé avec le produit. Plus plus d'informations, consulter l'article Exemples.

Test Des Déclencheurs

Test Des Déclencheurs

Quand la configuration du déclencheur est terminée, la moitié du travail est faite. Avant de déployer le déclencheur, vous devez convenablement tester les opérations qu'il doit effectuer en fonction des données d'entrée et vérifier l'exécution des actions.

Vous pouvez tester votre configuration pendant que vous continuez le développement Automation Builder. Certaines actions ont des capacités d'auto-test, ce qui vous permet de vous concentrer sur l'exécution de l'action individuelle. Vous pouvez également tester chaque déclencheur en lançant la commande Run Preview. Toutefois le test final doit toujours se faire dans un environnement réel, en fournissant des données réelles et utilisant des vrais déclencheurs, tout en surveillant l'exécution du déclencheur dans Automation Manager.

Test d'exécution des actions individuelles

Certaines actions ont une fonction d'aperçu, qui vous permet de changer les paramètres d'entrée et de voir le résultat de l'action à l'écran.

- **Utiliser un Filtre de Données.** L'action montrera un aperçu réel des données analysées. Les règles du filtre sélectionné sont appliquées au fichier de données d'entrée sélectionné et résultat affiché dans la table. Si vous utilisez des sous-zones ou zones d'assignation, vous pourrez voir l'aperçu pour chaque niveau de la définition du filtre.
- **Exécuter une Instruction SQL.** L'action affichera un aperçu de l'exécution de l'instruction SQL définie. Vous pourrez voir l'ensemble de données résultant de l'instruction SELECT et le nombre de rangées affectées par les instructions UPDATE, INSERT et DELETE. L'exécution de l'aperçu est une transaction sécurisée et vous pouvez restaurer tous les changements. Vous pouvez modifier les paramètres de requête introduits et voir comment ils influencent le résultat.
- **Web Service.** L'action affichera un aperçu d'exécution de la méthode sélectionnée (function) du Web Service. Vous pouvez modifier les paramètres d'entrée et voir comment ils influencent le résultat.
- **Exécuter le script.** L'action vérifiera les erreurs de syntaxe dans le script fourni, et l'exécutera. Vous pouvez modifier les paramètres d'entrée et voir comment ils influencent l'exécution du script.

Tester l'exécution du déclencheur et afficher un aperçu de l'étiquette à l'écran.

Pour tester le déclencheur complet, utilisez la fonction intégrée **Run Preview**. Vous pouvez effectuer un aperçu d'activation (run preview) pour chaque déclencheur, peu importe le type. Le déclencheur ne s'activera pas au changement de l'évènement surveillé, seul le déclencheur activé dans le Automation

Manager peut le faire. En fait, le déclencheur effectuera les actions en fonction des données sauvegardées dans un fichier. Vous devez vous assurer que vous avez un fichier contenant un échantillon de données qui seront acceptées par le déclencheur durant le déploiement en temps réel.

Le déclencheur exécutera toutes les actions définies, y compris le filtrage de données et l'aperçu de l'étiquette à l'écran. L'aperçu simulera le processus d'impression dans tous ses détails. Les étiquettes seront imprimées selon la même composition et contenu que l'aperçu. Cela inclut le nombre et le contenu des étiquettes. Cela vous informera du nombre de tâches d'impression créées et du nombre d'étiquettes dans chaque tâche ainsi qu'un aperçu de chaque étiquette. Vous pouvez naviguer d'une étiquette à l'autre dans la tâche d'impression sélectionnée.

Le volet Journal rapporte les mêmes informations que celles qui seraient affichées dans le Automation Manager. Développez les entrées du journal pour afficher tous les détails.

NOTE: Quand vous lancer l'aperçu d'impression, toutes les actions définies pour le déclencheur sélectionné seront activées, pas seulement l'action Impression d'étiquette. Soyez prudent quand vous utilisez des actions qui vont modifier les données, telles que Exécuter une instruction SQL (Exécuter l'Instruction SQL), ou Service Web, car leur exécution est irréversible.

Pour afficher un aperçu des étiquettes, effectuez les pas suivants :

1. Ouvrir la configuration du déclencheur.
2. Assurez-vous que la configuration du déclencheur est sauvegardée.
3. Cliquez le bouton **Afficher l'aperçu** dans le groupe Aperçu du ruban.
4. Naviguez vers le fichier de données fournissant le contenu typique que le déclencheur accep-tera.
5. Visualisez le résultat dans un onglet d'aperçu.

Tester le déploiement sur un serveur de pré-production

La meilleure pratique est de déployer la configuration vers Automation Manager sur un serveur de pré-production, avant le déploiement sur le serveur de production. Tester dans un environnement de pré-production permet d'identifier des problèmes de configuration qui n'ont pas été détectés durant les essais du déclencheur sur le Automation Builder seul. La performance peut aussi être mise à l'épreuve en ajoutant la charge sur le déclencheur et suivre ses résultats. Les tests fourniront d'importantes informations concernant la bande passante disponible et identifier les points faibles. En fonction des conclusions, vous pouvez ensuite implémenter diverse techniques d'optimisation, telles que l'optimisation de la conception d'étiquette pour réduire le flux d'impression et optimiser le flux général de données depuis l'application existante vers le NiceLabel Automation.

Différences importantes entre les tests réels et l'aperçu Automation Builder

Alors que l'aperçu du déclencheur à l'écran dans le Automation Builder procure une méthode rapide de test du déclencheur, vous ne devez pas uniquement vous y fier. Il peut y avoir des différences d'exécution entre l'aperçu et l'activation réelle du déclencheur quand vous utilisez Windows 64-bits.

Même si votre configuration fonctionne dans Automation Builder, assurez-vous de l'essayer en temps réel en utilisant le Service aussi.

- Quand vous lancez la commande **Run Preview** (lancer l'aperçu), la configuration l'exécutera dans Automation Builder, ce qui lance toujours l'application en 32-bits. L'aperçu de votre déclencheur dans Automation Builder lancera seulement l'exécution de test sur une plateforme 32-bits.
- Quand vous lancez le déclencheur en temps réel, la configuration l'exécutera dans Service, ce qui fonctionne comme application 32-bits sur Windows 32-bits, et fonctionnera comme une application 64-bit sur Windows 64-bit. Pour plus d'informations, consultez l'article Fonctionnement en Mode de Service.
- Les problèmes peuvent survenir quand le traitement du déclencheur est affecté par les différences de plateformes (32-bit vs 64-bit) :
  - **Accès à la Base de Données.** Les applications 64-bit nécessitent des pilotes de base de données 64-bit, et les applications 32-bit nécessitent des pilotes 32-bit. Pour lancer la configuration du Automation Builder et dans le Service, vous avez besoin de pilotes de base de données 32-bit et 64-bit pour accéder à votre base de données. Pour plus d'informations, consultez l'article Accéder aux bases de données.
  - **Syntaxe UNCpour les fichiers réseau.** Le compte de service ne peut pas accéder aux fichiers réseau partagés avec une lettre mappée. Vous devez utiliser la syntaxe UNC pour les fichiers réseau. Par exemple, utilisez `\\server\share\files\label.lbl` et pas `G:\files\label.lbl`, où G: est mappé vers `\\server\share`. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.
- Si votre Service NiceLabel Automation fonctionne sous un autre compte utilisateur que celui que vous utilisez pour Automation Builder, les comptes peuvent avoir des privilèges de sécurité différents. Si vous pouvez ouvrir l'étiquette dans le Automation Builder, le compte d'utilisateur pour le Service ne doit pas nécessairement pouvoir y accéder. Pour utiliser Automation Builder sous le même compte utilisateur que le Service, consultez Utiliser le même compte utilisateur pour configurer et exécuter les déclencheurs.

Protéger La Configuration Du Déclencheur Contre L'édition

La configuration du déclencheur peut être protégée en utilisant deux méthodes.

- **Verrouillage du déclencheur.** Cette méthode verrouille la configuration du déclencheur et la protège par un mot de passe. Personne ne peut éditer le déclencheur sans le mot de passe. Activer l'option **Verrouiller et encoder le déclencheur** dans déclencheur *Paramètres ->Sécurité*.
- **Paramétrage des permissions d'accès.** En utilisant cette méthode, vous dépendez des permissions d'utilisateur telles qu'elles sont définies dans les NiceLabel Automation Options. Vous pouvez activer des groupes d'utilisateurs et assigner différents rôles à chaque groupe. Si le groupe a reçu des droits d'édition, tous les membres du groupe pourront éditer les déclencheurs. Cette méthode requiert l'activation de la connexion d'utilisateur. Vous pouvez utiliser les utilisateurs Windows des groupes locaux ou d'active directory, ou vous pouvez définir les NiceLabel Automationutilisateurs. Voir **Droit et accès Utilisateur** dans Configuration.

Utilisation De La Couche De Transport Sécurisée (HTTPS)

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Enterprise**.

Vous pouvez votre trafic entrant sur le Déclencheur de Serveur HTTP et Déclencheur Web Service en activant le support HTTPS. HTTPS sécurise la transmission des messages échangés sur le réseau. La sécurité de communication utilise les certificats X.509 pour encoder les données circulant entre les éléments. Vos informations restent confidentielles aux yeux indiscrets car seul le client et le NiceLabel Automation peuvent décoder le trafic. Même si un utilisateur effectue une écoute clandestine sur la communication, il ne pourrait pas comprendre la signification des messages, car le trafic apparaît comme un flux d'octets aléatoires.

C'est une bonne pratique de sécurité d'encoder la communication dans les cas tels que :

- Vous travaillez avec des données est sensitives et confidentielles qui ne peuvent pas être exposés aux utilisateurs tierces.
- Le message doit passer au travers de réseaux qui sont en dehors de votre contrôle. Par exemple, ceci se passe quand vous envoyez des données à Automation par internet, et pas depuis le réseau local.

Activer la couche de transport sécurisée (HTTPS)

Effectuez les opérations suivantes pour activer la couche de transport sécurisée pour votre déclencheur.

Dans le système Windows :

1. Obtenez le certificat X.509 de l'éditeur de ceritificats digitaux (autorité de certificats - CA). Vous avez besoin d'un type de certificat pour 'authentification serveur'.

   NOTE: Si vous allez générer le certificat vous-même, assurez-vous d'importer le certificat CA dans le magasin de l'Autorité de Confiance, pour que la signature CA puisse être vérifiée avec le certificat du serveur.

2. Installez le certificat X.509 dans le système, sur lequel NiceLabel Automation est installé. Assurez-vous que le certificat est visible pour le compte d'utilisateur sous lequel fonctionne le service NiceLabel Automation. La meilleure pratique est d'installer le certificat dans le magasin de l'ordinateur local, pas dans le magasin de l'utilisateur actuel. Delà permettre au NiceLabel Automation d'utiliser le certificat, même s'il ne fonctionne pas avec le compte utilisateur sous lequel vous êtes connecté.

   1. Ouvrir une fenêtre de Commande.
   2. Tapez **mmc** et appuyez la touche ENTER (assurez-vous que vous disposez des privilèges administratifs).
   3. Dans le menu Fichier, cliquez **Ajouter/Enlever Enfichage**.
   4. Dans l'interface **Ajouter un Enfichage Autonome,** sélectionnez **Certificats**.
   5. Cliquez **Ajouter**.

6. Dans l'interface **Enfichage de Certificats**, sélectionnez **Compte ordinateur** et cliquez **Suivant**.
7. Dans l'interface **Sélectionner l'Ordinateur**, cliquez **Terminer**.
8. Dans l'interface **Ajouter/Enlever Enfichage**, cliquez **OK**.
9. Dans la fenêtre Racine de la Console, développez **Certificats>Personnel**.
10. Clic-droit sur le dossier Certificats et sélectionnez **Toutes tâches>Importer**.
11. Suivez les instructions pour importer le certificat.

3. Extraire l'emprunte d'un certificat que vous venez d'importer.
    1. En restant dans le MMC, effectuez un double-clic sur le certificat.
    2. Dans l'interface du **Certificat**, cliquez l'onglet **Détails**.
    3. Naviguez dans la liste de champs et cliquez sur Emprunte.
    4. Copiez les caractères hexadécimaux du champ. Enlevez les espaces entre les nombres hexadécimaux. Par exemple, l'emprunte "a9 09 50 2d d8 2a e4 14 33 e6 f8 38 86 b0 0d 42 77 a3 2a 7b" doit être spécifiée comme "a909502dd82ae41433e6f83886b00d4277a32a7b" dans le code. C'est le `HachCert` requis dans le pas suivant.
4. Lier le certificat à l'adresse IP et au port sur lequel le déclencheur fonctionne. Cette action va activer le certificat sur le numéro de port sélectionné.

    Ouvrez **Invite de Commande** (assurez-vous que vous l'utilisez avec les privilèges administrateur) et lancez la commande suivante :

    ```
    netsh http add sslcert ipport=0.0.0.0:56000
    certhash=7866c25377554ca0cb53bcdfd5ee23ce895bdfa2 appid={A6BF8805-1D22-42C2-
    9D74-3366EA463245}
    ```

    où :

    - `ipport` est la paire adresse IP-port, sur laquelle le déclencheur fonctionne. Laissez l'adresse IP à 0.0.0.0 (ordinateur local), mais changez le numéro de port pour qu'il corresponde au numéro de port dans la configuration du déclencheur.
    - `certhash` est l'emprunte (SHA hash) du certificat. Elle a une longueur de 20 octets et est spécifiée comme un chaîne hexadécimale.
    - `appid` est le GUID de l'application propriétaire. Ici vous pouvez utiliser n'importe quel GUID, même celui de l'exemple ci-dessus.

Dans la configuration du déclencheur :

1. Dans votre déclencheur HTTP ou Web Service, activez l'option **Connexion Sécurisée (HTTPS)**.
2. Rechargez la configuration dans le Automation Manager.

Désactiver la couche de transport sécurisée (HTTPS)

Dans le système Windows :

1. Déliez le certificat de la paire Adresse IP-port. Lancez la commande suivante dans l'Invite de Commande (assurez-vous que vous l'utilisez avec les privilèges administrateur) :

   ```
   netsh http delete sslcert ipport=0.0.0.0:56000
   ```

   où :

   - `ipport` est la paire adresse IP-port, sur laquelle le déclencheur fonctionne et à laquelle vous avez lié le certificat

Dans la configuration du déclencheur :

1. Dans votre déclencheur HTTP ou Web Service, désactivez l'option **Connexion Sécurisée (HTTPS)**.
2. Rechargez la configuration dans le Automation Manager.

Exécuter Et Gérer Les Déclencheurs

Déployer La Configuration

Quand vous avez configuré et testé les déclencheurs dans le Automation Builder, vous devez déployer la configuration vers le service NiceLabel Automation et démarrer les déclencheurs. A ce moment les déclencheurs prennent vie et commencent à surveiller les évènements définis.

Pour déployer la configuration, utilisez les méthodes suivantes.

Déployer de Automation Builder

1. Démarrer Automation Builder.
2. Charger la configuration.
3. Allez à l'onglet **Éléments de Configuration**.
4. Cliquez le bouton **Déployer la Configuration** dans le groupe du ruban Déployer.
   La configuration sera chargée dans le Automation Manager fonctionnant sur la même machine.
5. Démarrez les déclencheurs que vous voulez activer.

Si cette configuration était déjà chargée, le déploiement forcera le rechargement, gardant l'état actif des déclencheurs.

Déployer de Automation Manager

1. Démarrer Automation Manager.
2. Allez à l'onglet **Déclencheurs**.
3. Cliquez le bouton **+Ajouter** et naviguez vers la configuration sur le disque.
4. Démarrez les déclencheurs que vous voulez activer.

Déployer depuis la ligne de commande

Pour déployer la configuration `C:\Project\Configuration.MISX` et exécuter le déclencheur dans le déclencheur nommé `CSVTrigger`, effectuez les opérations suivantes :

```
NiceLabelAutomationManager.exe ADD c:\Project\Configuration.MISX
NiceLabelAutomationManager.exe START c:\Project\Configuration.MISX CSVTrigger
```

Pour plus d'informations, voir l'article Contrôler le Service avec les paramètres de ligne de Commande.

Options De Journalisation Des Évènements

NiceLabel Automation sauvegardera les évènements vers diverses destinations, selon leur scénario de déploiement. Les deux premiers éléments de journalisation sont disponibles avec chaque produit NiceLabel Automation.

- **Journalisation dans la base de données.** La sauvegarde dans le journal interne de la base de données est toujours active et sauvegarde tous les évènements avec tous les détails. Durant la visualisation des informations journalisées, vous pouvez utiliser un filtre pour afficher les évènements correspondant aux règles. Pour plus d'informations, consultez l'article Utilisation du journal d'évènements
. Les données sont sauvegardées dans la base de données SQLite. C'est un journal référentiel temporaire, les évènements sont enlevés de la base de données sur base hebdomadaire. L'intervalle de rangement est configurable dans les Options. Les données des anciens évènements seront effacés de la base de données, mais la base de données ne sera pas comprimée (vidée), elles peuvent donc encore occuper de la place sur le disque. Pour les comprimer, utilisez un logiciel de gestion SQLite d'une partie tierce.
- **Journalisation dans Windows Application Event Log.** Les évènements important sont sauvegardés dans le journal Windows Application Event Log dans le cas où le NiceLabel Automation n'a pas pu démarrer, ce qui vous procure une ressource secondaire pour les évènements journalisés.

Gestion Des Déclencheurs

L'application Automation Manager est la partie de gestion du logiciel NiceLabel Automation. Si vous utilisez Automation Builder pour configurer les déclencheurs, vous utiliserez Automation Manager pour les déployer et les exécuter dans l'environnement de production. L'application vous permet de charger des déclencheurs de différentes configurations, de voir leur état actuel, de les démarrer/arrêter et de voir les détails de leur exécution dans le fichier du journal.

Vous pouvez modifier l'affichage des configurations chargées et leurs déclencheurs. Le dernier affichage est mémorisé et sera appliqué au prochain lancement de Automation Manager. Quand vous activez l'affichage **par état**, les déclencheurs de toutes les configurations ouvertes qui sont dans cet état seront affichés en même temps. Quand vous activez l'affichage **par configuration**, les déclencheurs de la configurations sélectionnée seront affichés en même temps, sans tenir compte de leur état. L'état du déclencheur est affiché en code de couleur dans l'icône du déclencheur, permettant une identification plus aisée.

Les détails du déclencheur affiché vont changer en temps réel à la détection des évènements du déclencheur. Vous pouvez voir les informations telles que le nom du déclencheur, le type de déclencheur, combien d'évènements ont déjà été exécutés, combien d'erreurs ont été détectées et le temps écoulé depuis le dernier évènement. Quand vous passez la souris sur le nombre de déclencheurs exécutés, vous verrez le nombre d'évènements du déclencheur en attente d'exécution.

NOTE: La configuration chargée est en mémoire cache. Si vous modifiez la configuration dan le Automation Builder, le Automation Manager ne l'appliquera pas automatiquement. Pour appliquer le changement, vous devez recharger la configuration.

Charger la configuration

Pour charger la configuration, cliquez le bouton **+Ajouter** et naviguez vers le fichier de configuration (.MISX). Les déclencheurs de la configuration seront chargés en état suspendu. Vous devez démarrer les déclencheurs pour les activer. Pour plus d'informations, voir l'article Déployer la Configuration.

La liste de configurations chargées et l'état de chaque déclencheur est mémorisé. Si le serveur est redémarré pour une raison quelconque, le Service NiceLabel Automation rétablit l'état que le déclencheur avait avant le démarrage.

Rechargement et enlèvement de la Configuration

Quand vous mettez la configuration à jour et la sauvegardez dans Automation Builder, le changement ne sera pas appliqué automatiquement dans Automation Manager. Pour recharger la configuration, effectuez un clic-droit sur le nom de configuration, ensuite sélectionnez **Recharger la Configuration**. Tous les déclencheurs seront rechargés. Si le cache de fichier est activé, le rechargement va forcer la synchronisation de tous les fichiers utilisés par les déclencheurs.

Démarrer / arrêter les déclencheurs

Quand vous chargez les déclencheurs d'une configuration, leur état par défaut est arrêté. Pour démarrer le déclencheur, cliquez le bouton **Démarrer** dans la zone des déclencheurs. Pour arrêter le déclencheur, cliquez le bouton **Arrêter**. Vous pouvez sélectionner plusieurs déclencheurs pour la même configuration et les démarrer / arrêter tous simultanément.

Vous pouvez aussi contrôle le démarrage/arrêt depuis la ligne de commande. Pour plus d'informations, voir l'article Contrôler le Service avec les paramètres de ligne de Commande.

Gestion des conflits de déclencheurs

Les déclencheurs peuvent être en erreur à cause des situations suivantes. Vous ne pouvez pas démarrer un tel déclencheur avant d'avoir résolu le problème.

- **Erreur de configuration du déclencheur ou incomplète.** Dans ce cas, le déclencheur n'est pas configuré, les caractéristiques obligatoires ne sont pas définies ou les actions définies pour cette imprimante ne sont pas configurées. Vous ne pouvez pas démarrer un tel déclencheur.
- **La configuration du déclencheur se croise avec un autre déclencheur.** Deux déclencheurs ne peuvent pas surveiller le même évènement.

> **EXEMPLE:** Deux déclencheurs de fichier ne peuvent pas surveiller le même fichier, deux déclencheurs HTTP ne peuvent pas accepter des données sur le même port. Si la configuration du déclencheur se croise avec un autre déclencheur, le second déclencheur ne fonctionnera pas, car l'évènement est déjà capturé par le premier déclencheur. pour plus d'informations, consultez l'onglet Journal de ce déclencheur.

Réarmement de l'état d'erreur

Quand l'exécution du déclencheur cause une erreur, l'icône du déclencheur change en couleur rouge, le déclencheur a un état d'erreur et les détails de l'évènement sont sauvegardés dans le journal de la base de données. Même si les évènements suivants se terminent avec succès, le déclencheur reste en état d'erreur jusqu'à ce que vous confirmez que vous avez compris l'erreur et que vous voulez annuler l'état. Pour confirmer l'erreur, cliquez sur l'icône à côté du compteur d'erreurs dans les détails du déclencheur.

Utilisation du panneau de notifications

Le panneau de notifications est la zone située au-dessus de la liste de déclencheurs dans l'onglet Déclencheurs, où les messages importants seront affichés. La zone de notifications affichera les

messages d'état, tels que "Mode d'essai" ou "Mode d'essai expiré", ou les **messages d'avertissement,** tels que "Traçage activé".

Visualisation des données du journal

Chaque activité du déclencheur est sauvegardée dans la base de données, y compris les évènements démarrage/arrêt du déclencheur, l'exécution avec succès des actions et les erreurs rencontrées durant l'exécution. Cliquez sur le bouton Journal pour visualiser les évènements sauvegardés pour le déclencheur sélectionné. Pour plus d'informations, consultez l'article Utilisation du journal d'évènements.

Utilisation Du Journal D'évènements

Toutes les activités du logiciel NiceLabel Automation sont sauvegardées dans la base de données pour l'historique et la résolution de problèmes. Quand vous cliquez le bouton **Journal** dans l'onglet Déclencheurs, les évènements du déclencheur particulier sont affichés. Le journal affichera les informations pour tous les évènements correspondant au filtre défini.

Les données du journal sont utiles pour la résolution de problèmes. Si l'action du déclencheur ne peut pas être exécutée, l'application sauvegarde une description de l'erreur dans le fichier du journal, ce qui vous aidera à résoudre le problème.

NOTE: La rétention de données par défaut est de 7 jours et est configurable dans la Configuration. Pour minimaliser la taille de la base de données du journal sur les systèmes chargés, vous pouvez diminuer la période de rétention. Voir options **NiceLabel Automation Configuration**.

Filtrage des évènements

Les filtres configurables :

- **Configuration et déclencheurs.** Spécifie quels évènements à afficher, les évènements du déclencheur sélectionné ou les évènements de tous les déclencheurs de la configuration sélectionnée.
- **Période sauvegardée.** Spécifie la période de temps durant laquelle les évènements se sont déroulés. Par défaut **Dernières 5 minutes**.
- **Niveau de l'évènement.** Spécifie le type (l'importance) des évènements que vous voulez afficher. **Erreur** est un type d'évènement qui va interrompre l'exécution. **Avertissement** est un type d'évènement pour lequel une erreur apparaît mais est configuré pour l'ignorer. **Information** est un type d'évènement qui sauvegarde toutes les informations non-erronées. Le niveau du journal est configurable dans **NiceLabel Automation Configuration**.
- **Filtrer par texte.** Vous pouvez afficher tous les évènements qui contiennent la chaîne de caractères fournie. Utilisez cette option pour la résolution d'erreurs des déclencheurs chargés. Le filtre sera appliqué au champ de description du déclencheur.

Effacement de la base de données du journal

Vous pouvez effacer le journal dans Automation Builder. Pour effacer la base de données du journal, cliquez le bouton **Effacer le journal**.

AVERTISSEMENT Utilisez avec précautions, car il n'y a pas de retour possible. Ceci enlèvera **TOUS** les évènements sauvegardés dans la base de données du journal, et est appliqué à tous les déclencheurs, pas seulement le déclencheur actuel.

Performances Et Options De Feedback

Traitement Parallèle

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

La ligne de produits NiceLabel Automation a été développée pour permettre le traitement parallèle des processus entrants et sortants. Cela assure une efficacité maximale de tout système sur lequel le logiciel a été installé. NiceLabel Automation peut effectuer plusieurs tâches simultanément, tout en conservant l'ordre dans lequel les déclencheurs sont arrivés. La rapidité des tâches d'impression est grandement dépendante du matériel sur lequel le logiciel est implémenté.

Traitement parallèle d'entrée

Vous pouvez exécuter plusieurs déclencheurs sur la même machine et ils répondront tous aux changements surveillés dans les évènements simultanément. Chaque déclencheur mémorise les données des évènements non-traités de la liste dans la file. La liste va mettre les données entrantes en mémoire tampon dans le cas où aucune des tâches d'impression n'est actuellement disponible. Dès qu'une tâche d'impression est disponible, la première tâche est prise de la file en utilisant le principe FIFO (Premier entré, Premier sorti). Cela permet le respect d'impression correct des données d'entrée. Ceci n'assure toutefois pas le principe FIFO pour l'impression. Voir paragraphe suivant.

NOTE: Non seulement vous pouvez exécuter plusieurs déclencheurs en parallèle. Chaque déclencheur peut aussi permettre des connexions concurrentes. Le déclencheurs TCP/IP, HTTP, et Web Service acceptent tous les connexions concurrentes de plusieurs clients. De plus, le déclencheur à fichier peut être configuré pour surveiller un ensemble de fichiers dans un dossier, configurable par un masque.

Traitement parallèle de Sortie

Généralement le résultat du déclencheur est le processus d'impression de l'étiquette. Vous voulez utiliser les données reçues et les imprimer sur les étiquettes. Le service NiceLabel Automation exécute les processus d'impression (c.-à-d. "moteurs d'impression") en parallèle en arrière-plan. Les processeurs modernes ont deux ou plusieurs processeurs centraux d'exécution appelées "cœurs". Les processeurs multiples peuvent exécuter des instructions multiples en même temps, ce qui augmente la vitesse totale de traitement, dans le cas du NiceLabel Automation, ils vont augmenter la vitesse de traitement des tâches d'impression, et donc améliorer les performances d'impression.

Par défaut, chaque produit NiceLabel Automation va exécuter les tâches d'impression dans processus séparé sur chaque cœurs qui est disponible dans la machine. Au plus votre processeur central est puissant, au plus la capacité de traitement est élevée. Ceci optimise l'usage de la puissance disponible du processeur central. Durant l'installation, le logiciel installe des valeurs par défaut raisonnables pour que chaque cœur disponible traite un processus d'impression et en circonstances normales, vous ne devrez jamais effectuer de changements. Si vous devez effectuer un changement, consultez l'article Changer les paramètres par défaut d'Impressions Multiples.

Quand vous disposez d'une grande quantité de processus d'impression, les données du premier évènement peut être imprimé par un processus d'impression, alors que les données du second évènement peuvent être imprimées simultanément par un processus d'impression différent, si un deuxième processus d'impression est disponible à ce moment. Si le second évènement n'a pas fourni beaucoup de données, le processus d'impression peut fournir les données pour l'imprimante plus rapidement que le premier processus d'impression, rompant l'ordre. Dans ce cas, les données du deuxième évènement pourraient être imprimées avant les données du premier évènement. Pour aussi assurer le principe FIFO pour l'impression, consultez l'article Mode d'impression Synchrone.

Mise En Cache De Fichiers

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Enterprise**.

Pour améliorer le temps de la première étiquette et les performances générales, NiceLabel Automation supporte la mise en cache de fichiers. Vous pouvez rencontrer des délais d'impression quand vous chargez les étiquettes, images et bases de données depuis vos réseaux partagés. NiceLabel Automation doit extraire tous les fichiers nécessaires avant de pouvoir démarrer le processus d'impression.

Il y a deux nivaux complémentaires de mise en cache.

- **Cache mémoire.** Le cache mémoire consiste à conserver les fichiers déjà utilisés en mémoire. Les étiquettes qui ont été utilisées au moins une fois sont chargées en cache mémoire. Quand le déclencheur requiert l'impression de la même étiquette, l'étiquette est immédiatement disponible pour le processus d'impression. Le cache mémoire est activé par défaut. Le contenu du cache mémoire sera vidé pour une configuration particulière, quand vous enlevez ou rechargez cette configuration. Le fichier d'étiquette est contrôlé en changements pour chaque action Ouverture Étiquette. Si une étiquette plus récente est disponible, elle sera chargée automatiquement, remplaçant l'ancienne version en cache.
- **Cache Persistant.** Le cache persistant sauvegarde les données sur le disque et est destiné au stockage intermédiaire de fichiers. Le cache est géré par l'objet fichier. Quand un fichier est requis par le partage réseau, le service commence par vérifier si le fichier est déjà présent en cache et l'utilise. Si le fichier n'est pas en cache, il sera extrait du partage réseau et mis en cache pour une utilisation ultérieure. Le service de mise en cache met continuellement à jour le contenu du cache avec les versions plus récentes des fichiers. Vous pouvez configurer les intervalles de temps pour les contrôles de changement de version dans la Configuration de NiceLabel Automation.

  NOTE: Quand vous rechargez la configuration, le service de mise en cache va vérifier si une version plus récente du fichier existe sue le réseau partagé et mettre à jour le cache local.

Activation du Cache Persistant

Pour activer et configurer le cache persistant, ouvrez la Configuration du NiceLabel Automation, sélectionnez NiceLabel Automation Paramètres et activer **Cache fichiers distants**.

- **Mise à jour des fichiers cache.** Définit l'intervalle de temps en minutes durant lequel le cache sera synchronisé avec les fichiers dans leur dossier d'origine. Ceci est l'intervalle de temps durant lequel vous autorisez le système à utiliser l'ancienne version du fichier.
- **Supprimer les fichiers cache quand ils sont plus âgés que.** Définit l'intervalle de temps en jours qui sera utilisé pour éliminer tous les fichiers du cache qui n'ont pas été utilisés durant cette période.

NiceLabel Automation utilisera le dossier local suivant comme cache pour les fichiers distants :

```
%ProgramData%\EuroPlus\NiceLabel Automation\system.net\FileCache
```

Forçage de Rechargement du contenu Cache

NiceLabel Automation va mettre à jour automatiquement le contenu du cache à l'intervalle de temps défini (la valeur par défaut est de 5 minutes).

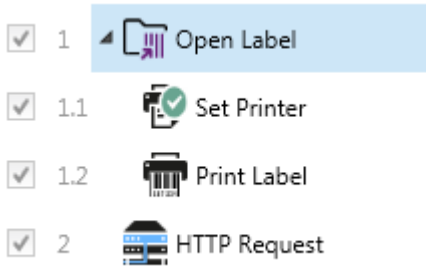
Pour forcer le rechargement du cache manuellement, effectuez les opérations suivantes :

1. Ouvrez Automation Manager.
2. Localisez la configuration qui contient le déclencheur pour lequel vous voulez forcer le rechargement des étiquettes.
3. Effectuez un clic-droit sur la configuration.
4. Sélectionnez **Recharger la Configuration**.

Traitement D'Erreurs

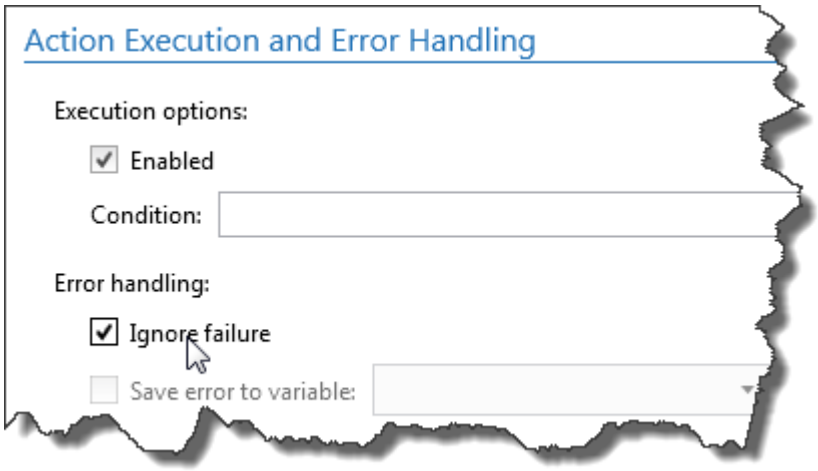
Quand une erreur survient durant l'exécution d'une action; NiceLabel Automation arrête l'exécution de toutes les actions dans le déclencheur. Si vous avez des actions définies après l'action actuelle, elles ne seront pas exécutées.

Par exemple, les actions sont définies comme dans la copie d'écran. Si l'action **Définir l'imprimante** échoue, à cause du nom erroné ou l'imprimante est inaccessible, les actions **Imprimer l'étiquette** et **Requête HTTP** ne seront pas exécutées. Le traitement de l'action s'arrête a**Définir l'imprimante**, Automation Manager affichera que le déclencheur est en état d'erreur et le retour du déclencheur (s'il est activé) sera en termes de "mauvaise sélection d'imprimante / imprimante inaccessible".



Toutefois, dans ce cas particulier, vous ne voulez pas utiliser le retour asynchrone (envoyé automatiquement quand il est activé dans le déclencheur supportant le retour synchrone). Le retour d'état doit être fourni de façon asynchrone avec l'action **Requête HTTP** après la création de la tâche d'impression (ou pas). Après la fin du processus d'impression, vous voulez mettre une application à jour avec son état. Vous enverrez un message en format HTTP vers cette application.

Dans ce cas, l'action **Requête HTTP** doit être exécutée, sans tenir compte du succès de toutes les actions situées au-dessus de celle-ci dans la liste. Vous devez activer l'option **Ignorer les échecs** pour toutes les actions qui sont au-dessus de l'action **Requête HTTP**. L'option est disponible dans les options des actions Exécution et traitement des erreurs.



Si une action particulière échoue, NiceLabel Automation commencera l'exécution de l'action suivante dans le niveau précédent de la hiérarchie.

Par exemple, si l'action **Définir l'imprimante** dans le niveau 1.1 échoue, l'exécution ne continuera pas avec l'action **Imprimer l'étiquette** du niveau 1.2 car il échouera probablement aussi, mais va continuer l'action **Requête HTTP** du niveau 2, car c'est l'action suivante dans le niveau supérieur de la hiérarchie.

Pour plus d'information, consultez les articles Feedback de la tâche d'impression et Mode d'impression Synchrone.

Mode D'impression Synchrone

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

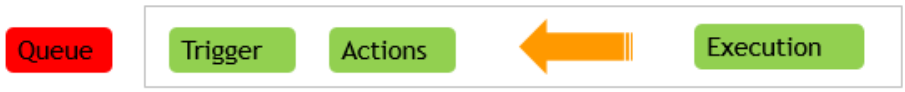
Mode d'impression Asynchrone

NiceLabel AutomationLe mode d'opération par défaut est asynchrone. C'est une forme d'impression, dans laquelle un déclencheur envoie les données d'impression et ensuite ferme la connexion vers le sous-système d'impression. Le déclencheur n'attend pas les résultat du processus d'impression et ne reçoit pas de feedback. Dès que les données ont été envoyées, le déclencheur est prêt à recevoir un nouveau flux de données. Le mode asynchrone amplifie les performances du déclencheur et augmente le nombre de déclencheurs qui peuvent être traités par période de temps. Chaque processus d'impression a une mémoire tampon devant lui, dans laquelle le déclencheur envoie les requêtes d'impression. La mémoire tampon absorbe les pointes et assure que les données ne sont pas perdues.

Si l'erreur survient durant le traitement, elle sera enregistrée dans Automation Manager (et NiceLabel Control Center, si vous l'utilisez), mais le déclencheur n'en est pas informé. Quand vous utilisez le mode d'impression asynchrone, vous ne pouvez pas définir l'exécution d'actions conditionnelles, si l'exécution du déclencheur est en erreur.

Mode d'impression Synchrone

Au contraire, le mode synchrone ne coupe pas la connexion avec le processus d'impression. Dans ce mode, le déclencheur envoie les données d'impression et conserve la connexion établie avec le sous-système d'impression tant qu'il traite les actions. Quand le processus d'impression se termine (avec succès ou avec une erreur), le déclencheur reçoit le feedback concernant l'état. Vous pouvez utiliser cette information pour les actions qui sont définies dans le même déclencheur et prendre la décision d'exécuter une autre action si une erreur survient. Vous pouvez aussi renvoyer l'état de la tâche d'impression à l'application qui a fourni les données. Pour plus d'informations, consultez l'article Feedback de la tâche d'impression.



EXEMPLE: Vous pouvez rapporter l'état d'impression à l'application ERP qui a fourni les données.

Le mode synchrone sera utilisé quand vous voulez recevoir le feedback d'état dans le déclencheur, ou quand vous voulez assurer le mode d'impression FIFO (les données reçues dans les évènements du déclencheur sont imprimées dans le même ordre qu'elles sont reçues).

NOTE: Quand le déclencheur fonctionne en mode d'impression synchrone, il communique avec un seul processus d'impression. L'activation du mode d'impression synchrone assure la méthode FIFO de manipulation des évènements dans la direction de sortie (impression). Le traitement en mode multi-cœur par défaut ne peut pas assurer l'ordre d'impression.

Activer le mode d'impression Synchrone

Le mode synchrone est définissable par déclencheur. Pour activer le mode synchrone dans un déclencheur, effectuez les opérations suivantes :

1. Ouvrez les caractéristiques du déclencheur.
2. Allez à l'onglet **Paramètres**.
3. Sélectionnez l'option **Autre**.
4. Dans la section **Feedback du moteur d'impression**, activez l'option **Impression Supervisée**.

Feedback De La Tâche D'impression

Les fonctionnalités de cet élément ne sont pas toutes disponibles dans chaque produit NiceLabel Automation.

L'application qui fournit les données d'impression au NiceLabel Automation peut s'attendre à recevoir des informations concernant l'état des tâches d'impression. Le retour peut être simple tel que "Tout OK" dans le cas de la génération réussie d'une tâche d'impression, ou plu détaillée en cas de problème. Pour des raisons de performances, par défaut NiceLabel Automation désactive les possibilités de feedback. Ceci assure l'exécution rapide de l'impression car le déclencheur ne s'occupe pas du processus d'exécution d'impression. Les erreurs seront enregistrées dans le journal de la base de données, mais le déclencheur ne va pas les traiter.

Vous pouvez également utiliser cette méthode pour envoyer un feedback concernant les autres données que le déclencheur peut collecter, tel que l'état des imprimantes réseau, le nombre de tâches d'impression dans le spouleur, la liste d'étiquettes dans un dossier, la liste de variables dans le fichier d'étiquette spécifié et beaucoup d'autres.

NOTE: Pour activer le support de feedback pour le moteur d'impression, vous devez activer le mode d'impression synchrone. Pour plus d'informations, Consultez l'article Mode d'impression Synchrone.

Vous pouvez fournir le retour d'état de deux façons.

Le déclencheur fournit le feedback d'état des tâches d'impression (feedback Synchrone)

Certains déclencheurs ont la possibilité de feedback incorporée. Quand le mode d'impression synchrone est activé, le déclencheur connait en interne l'état de la tâche d'impression. Le client peut envoyer les données au déclencheur, garder la connexion ouverte et attendre le retour. Pour utiliser cette méthode, vous devez avoir un déclencheur qui la supporte.

Quand l'erreur se présente dans une des actions, la variable interne `LastActionErrorDesc` contiendra le message d'erreur détaillé. Vous pouvez envoyer sa valeur telle-quelle ou la personnaliser.

Pour plus d'informations, consultez les détails du déclencheur respectif.

- Déclencheur Web Service. Ce déclencheur supporte le feedback par défaut. Le document WSDL (Web Service Description Language) décrit les détails concernant l'interface Service Web et comment activer le feedback. Vous pouvez utiliser la réponse par défaut qui enverra la description d'erreur en cas de défaillance de l'action d'impression. Ou vous pouvez personnaliser la réponse et renvoyer le contenu d'une variable. La variable peut contenir des données diverses, y compris un aperçu de l'étiquette ou la tâche d'impression (données binaires).
- Déclencheur de Serveur HTTP. Ce déclencheur supporte le feedback par défaut. NiceLabel Automation va utiliser le code de réponse HTTP standard pour indiquer l'état de la tâche d'impression. Vous pouvez personnaliser la réponse HTTP et renvoyer le contenu d'une variable. La variable peut contenir des données diverses, y compris un aperçu de l'étiquette ou la tâche d'impression (données binaires).
- Déclencheur Serveur TCP/IP. Ce déclencheur supporte le feedback, mais pas automatiquement. Dans ce cas, vous devez configurer le client qui fournit les données pour ne pas interrompre la connexion après l'envoi des données. Après la fin du processus d'impression, l'action suivante dans la liste peut être Envoyer les données au port TCP/IP avec le paramètre **Répondre à l'expéditeur.** Vous pouvez envoyer le feedback sur la connexion restée ouverte.

L'action fournit le feedback d'état de la tâche d'impression (feedback Asynchrone)

Pour les déclencheurs qui ne supportent pas le feedback en interne ou si vous voulez envoyer des

messages de feedback durant le traitement du déclencheur, vous pouvez définir une action qui renverra un feedback à une certaine destination. Dans ce cas, l'application fournissant les données peut fermer la connexion dès que les données pour le déclencheur ont été fournies.

> **EXEMPLE:** Vous avez utilisé un déclencheur TCP/IP pour capturer les données. Le client a terminé la connexion immédiatement après l'envoi des données, donc nous ne pouvons pas répondre sur la même connexion. Dans ce cas, vous pouvez utiliser un autre canal pour renvoyer le feedback. Vous pouvez configurer une des actions de connexion sortante, telle que Exécuter une instruction SQL, Ouvrir un Document / Programme, Requête HTTP, Envoyer les données au port TCP/IP et autres. Vous devez remplacer une telle action dans l'action Impression d'étiquette.

Si vous voulez envoyer un feedback pour un statut spécifique, tel que "erreur survenue", vous pouvez utiliser une des méthodes suivantes.

- **Utiliser une condition sur l'action.** La tâche d'impression est exposée dans deux variables internes (`LastActionErrorID` et `LastActionErrorDesc`). La première contient l'ID de l'erreur ou contiendra la valeur `0` s'il n'y a pas d'erreur. La seconde contient un message d'erreur détaillé. Vous pouvez utiliser les valeurs de ces variables dans des conditions sur les actions que vous voulez exécuter en cas d'erreur. Par exemple, vous voulez utiliser l'action **HTTP Request** après l'impression et envoyer le feedback seulement quand une erreur survient. You would do the following:

  1. Open trigger properties.
  2. Dans le groupe Variable du ruban, cliquez le bouton **Variables Internes** et activez la variable `LastActionErrorID`.
  3. Allez à l'onglet Actions.
  4. Ajoutez l'action **Envoyer les Données au HTTP**.
  5. Dans les caractéristiques de l'action, développez **Afficher les options d'exécution et de traitement d'erreurs** .
  6. Pour **Condition**, introduisez comme suit. L'action ayant cette condition s'exécutera seulement si l'erreur survient et `LastErrorActionID` contient l'ID de l'erreur (toute valeur plus grande que `0`). Par défaut, les conditions s'exécutent en utilisant la syntaxe VB Script.

     ```
     LastErrorActionID > 0
     ```

     Vous devrez également activer l'option **Ignorer l'erreur** pour chaque action dont vous prévoyez l'échec. Cela préviendra Automation de ne pas arrêter complètement l'exécution des actions, mais de continuer par l'action suivante dans le même niveau hiérarchique.

     > **NOTE:** Pour plus d'informations, consultez l'article Traitement d'Erreurs
  7. .

- **Utiliser l'action Essai.** L'action Essai élimine la nécessité d'encoder les conditions. L'action vous fournit deux espaces réservés. L'espace réservé **Do** (faire) va contenir les actions que vous voulez exécuter. Si une erreur survient, l'exécution s'interrompt et les actions dans l'espace réservé **Pour erreur** seront exécutées. Les actions de connectivité-de-sortie de cet espace réservé sont

utilisées pour procurer le feedback du statut d'une tâche d'impression. Pour plus d'informations, consultez l'article Essai.

Cluster Haute Disponibilité (Fail-over)

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Enterprise**.

NiceLabel Automation supporte le cluster Microsoft haute-disponibilité (fail-over). Un cluster à reprise (failover) est un groupe d'ordinateurs indépendants qui travaillent ensemble pour augmenter la disponibilité d'impression d'étiquettes.NiceLabel Automation Les serveurs groupés (appelés nœuds) sont raccordés physiquement par des câbles et par un logiciel. Si un ou plusieurs des nœuds d'ensembles est défaillant, les autres nœuds reprennent le service (ce processus est connu sous le nom de fail-over). De plus, les rôles repris sont surveillés de façon pro-active pour vérifier qu'ils fonctionnent convenablement. S'ils ne fonctionnent pas, ils sont redémarrés ou déplacés vers un autre nœud. Les client qui envoient des données se connecteront à l'adresse IP appartenant au cluster, pas à l'adresse IP du nœud.

Pour activer le NiceLabel Automation en Haute-disponibilité, effectuez les opérations suivantes :

- Activez l'élément Microsoft Fail-over Clustering dans vos Serveurs Windows.
- Installez NiceLabel Automationsur chaque nœud.
- Activez le support de reprise en nœud (cluster fail-over) dans les caractéristiques NiceLabel Automation de chaque nœud.
  Effectuez :
  1. Ouvrir **NiceLabel Automation Configuration.**
  2. Sélectionnez la section **Support d'Ensembles** (Cluster Support).
  3. Activer le **Support de Reprise en Nœud** (Failover Cluster Support).
  4. Naviguez vers le dossier, situé en dehors des deux nœuds, mais toujours accessible avec les pleins droits d'accès au logiciel NiceLabel Automation. Les fichiers importants du système que les deux nœuds nécessitent seront des copies vers ce dossier.
- Configurez l'Ensemble (cluster) pour qu'il démarre NiceLabel Automation au 2e nœud si le nœud Maître est arrêté.

Cluster Répartition De Charge

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Enterprise**.

NiceLabel Automation supporte le Cluster répartiteur de charge Microsoft (load-balancing cluster). Un cluster à répartition de charge est un groupe d'ordinateurs indépendants qui travaillent ensemble pour augmenter la haute-disponibilité d'impression d'étiquettes au travers de NiceLabel Automation. Les serveurs groupés (appelés nœuds) sont raccordés physiquement par des câbles et par un logiciel. Les requêtes entrantes d'impression d'étiquettes sont distribuées entre touts les nœuds dans un cluster. Les client qui envoient des données se connecteront à l'adresse IP appartenant au cluster, pas à l'adresse IP du nœud.

NOTE: Vous pouvez utiliser les déclencheurs sur base TCP/IP avec cluster répartiteur de charge, ceci inclut Déclencheur Serveur TCP/IP, Déclencheur de Serveur HTTP et Déclencheur Web Service.

Pour activer la répartition de charge du NiceLabel Automation, effectuez les opérations suivantes :

- Activez l'élément Microsoft Load-balancing Clustering dans vos Serveurs Windows.
- Installez NiceLabel Automationsur chaque nœud.
- Chargez les même fichiers de configuration dans Automation Manager sur chaque nœud.

Comprendre Les Structures De Données

Comprendre Les Structures De Données

Ce chapitre démontre la structure de donnée basique qui est fréquemment utilisée dans les scénarios d'automation. Nous devons lire les structures, extraire les valeurs d'intérêt et les imprimer sur l'étiquette. Chaque échantillon mentionné est utilisé dans les configurations d'échantillons qui sont installées avec le logiciel. Plus plus d'informations, consultez l'article Exemples.

- Base de données de Texte
- CSV Composé
- Fichiers Binaires
- Données existantes
- Fichiers de Commande
- Données XML

Fichiers Binaires

Les fichiers binaires sont des fichiers qui ne contiennent pas seulement du texte mais contiennent des caractères binaires tels que les codes de contrôle (caractères sous le code ASCII 32). Le Configuration de Filtre de Données Non-structurées supporte les caractères binaires. Vous pouvez utilisez des caractères binaires pour définir des positions de champs, et vous pouvez aussi utiliser les caractères binaires pour les valeurs de champs.

Un exemple typique serait l'exportation de données d'un système existant, sur lequel les données d'étiquettes sont délimitées par un caractère <FF> - saut de page.

Exemple

Dans ce cas; le déclencheur capture le flux d'impression. La section surlignée en jaune doit être extraite du flux et envoyée à une imprimante différente. Le filtre est configuré pour rechercher le <FF> en position de fin de champ.

```
<ESC>%-12345X@PJL USTATUSOFF
@PJL INFO STATUS
@PJL USTATUS DEVICE=ON
<ESC>%-12345X<ESC>%-12345X
```

```
^^02^L
 ^^02^O0270
 D11
 H15
 PE
 SE
 Q0001
 131100000300070001-001-001
 1e42055007500500001001019
 1322000001502859
 W
 E
<FF><ESC>%-12345X<ESC>%-12345X@PJL USTATUSOFF
<ESC>%-12345X
```

Plus plus d'informations, consultez l'article Exemples.

Fichiers De Commande

Les fichiers de Commande sont des fichiers texte contenant des commandes qui seront exécutées une par une, de haut en bas. NiceLabel Automation supporte les fichiers de commande originaux, ainsi que les fichiers de commande XML Oracle et SAP. Pour plus d'informations, voir les articles Référence et Résolution de Problèmes, Caractéristiques Oracle XML et Caractéristiques SAP AII XML.

Exemple

L'étiquette `label2.lbl` sera imprimée sur l'imprimante `CAB A3 203DPI`.

```
LABEL "label2.lbl"
SET code="12345"
SET article="FUSILLI"
SET ean="383860026501"
SET poids="1,0 kg"
PRINTER "CAB A3 203DPI"
PRINT 1
```

Plus plus d'informations, consultez l'article Exemples.

CSV Composé

Compound CSV (combiné) est un fichier texte contenant une structure CSV ainsi qu'une entête multi-ligne dans une autre structure. Le contenu ne peut pas être analysé par un seul filtre. Vous devez configurer deux filtres, le premier un Configuration de Filtre de Texte Structuré pour les champs dans la section CSV et un second Configuration de Filtre de Données Non-structurées pour les champs dans la section d'entête. Vous définirez deux actions Utilisation d'un Filtre de Données dans les actions et exécuterez les deux filtres sur les données reçues.

Exemple

Les données de la ligne 3 jusqu'à la fin du document contient une structure CSV et sont analysées par le Filtre de Texte Structuré. Les données dans les 2 premières lignes n'ont pas de structure particulière et sont analysées par le Filtre de Texte non-Structuré.

```
OPTPEPPQPF0 NL004002 ;F75-TEP77319022891-001-001
OPT2 zg2lbprt.p 34.1.7.7 GOLF+ label print
"printer";"label";"lbl_qty";"f_logo";"f_field_1";"f_field_2";"f_field_3"
"Production01";"label.lbl";"1";"logo-nicelabel.png";"ABCS1161P";"Post: ";"1"
 "Production01";"label.lbl";"1";"logo-nicelabel.png";"ABCS1162P";"Post: ";"2"
 "Production01";"label.lbl";"1";"logo-nicelabel.png";"ABCS1163P";"Post: ";"3"
 "Production01";"label.lbl";"1";"logo-nicelabel.png";"ABCS1164P";"Post: ";"4"
 "Production01";"label.lbl";"1";"logo-nicelabel.png";"ABCS1165P";"Post: ";"5"
```

Plus plus d'informations, consultez l'article Exemples.

Données Existantes

Les données existantes sont une exportation non-structurée ou semi-structurée des applications existantes. Ceci n'est pas une structure de données CSV ou XML, vous devez donc utiliser le Configuration de Filtre de Données Non-structurées et définir les positions des champs d'intérêt. Le filtre va extraire les valeurs de champ pour que vous puissiez les imprimer sur les étiquettes.

Exemple

Il n'y a pas de règles de structure. Chaque champ doit être configuré manuellement.

```
HAWLEY ANNIE ER12345678 ABC XYZ
9876543210
PRE OP 07/11/12 F 27/06/47 Hopital St Georges


G015 134 557 564 9 A- 08/11/12 LDBS F- PB 1
G015 134 654 234 0 A- 08/11/12 LDBS F- PB 2
G015 134 324 563 C A- 08/11/12 LDBS F- PB 3









Dépistage Anti-corps : Négatif
Stockage de l'échantillon :
ÉCHANTILLON VALABLE 24 HEURES, PAS D'HISTORIQUE DE TRANSFUSION FOURNI


07/11/12 B,31.0001245.E O Rh(D) Pos PHO
RLUH BT
```

Plus plus d'informations, consultez l'article Exemples.

Base De Données De Texte

La base de donnée texte est un alias donné aux fichiers texte à champs structurés, tels que CSV (fichier à séparation par virgule), ou fichier à champs de largeur fixe. Dans chaque cas, vous pouvez cliquer le bouton **Importer la Structure de Données** et suivre les instructions de l'interface pour importer les champs. Si vous avez un fichier de données avec une structure délimitée et le nombre de champs varie d'une copie à l'autre, vous pouvez activer l'élément **Structure Dynamique** et laisser NiceLabel Automation traiter l'extraction de données et le mappage vers les variables automatiquement. Pour plus d'informations, voir l'article Activer la Structure Dynamique.

Exemple

- **Fichier à champs délimités.** La première ligne du fichier peut contenir les noms de champs que le filtre peut importer.

```
Product_ID;Code_EAN;Product_desc;Package
CAS006;8021228110014;CASONCELLI ALLA CARNE 250G;6
PAS501;8021228310001;BIGOLI 250G;6
PAS502GI;8021228310018;TAGLIATELLE 250G;6
PAS503GI;8021228310025;TAGLIOLINI 250G;6
PAS504;8021228310032;CAPELLI D'ANGELO 250G;6
```

- **Fichier à largeur de champs fixe.**

```
CAS006 8021228110014 CASONCELLI ALLA CARNE 250G 6
PAS501 8021228310001 BIGOLI 250G 6
PAS502GI 8021228310018 TAGLIATELLE 250G 6
PAS503GI 8021228310025 TAGLIOLINI 250G 6
PAS504 8021228310032 CAPELLI D'ANGELO 250G 6
```

Plus plus d'informations, consultez l'article Exemples.

Données XML

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

XML signifie eXtensible Markup Language. Les balises XML ne sont pas prédéfinies, vous pouvez librement définir vos propres balises qui décriront vos données. XML est conçu pour être auto-descriptif.

La structure XML est définie par des éléments, des attributs (et leurs valeurs), et texte (élément de texte).

Exemples

Oracle XML

Le traitement d'Oracle XML est intégré dans le logiciel. Vous ne devez pas configurer de filtres pour extraire les données, il suffit de lancer l'action intégrée Utiliser un fichier de Commande Oracle XML. Pour plus d'informations concernant la structure XML, voir l'article Caractéristiques Oracle XML.

```
<?xml version="1.0" standalone="no"?>
<labels _FORMAT="case.lbl" _PRINTERNAME="Production01" _QUANTITY="1">
<Label>
<variable name="CASEID">0000000123</variable>
<variable name="CARTONTYPE"/>
<variable name="ORDERKEY">0000000534</variable>
<variable name="BUYERPO"/>
<variable name="ROUTE"></variable>
<variable name="CONTAINERDETAILID">0000004212</variable>
<variable name="SERIALREFERENCE">0</variable>
<variable name="FILTERVALUE">0</variable>
<variable name="INDICATORDIGIT">0</variable>
<variable name="DATE">11/19/2012 10:59:03</variable>
</label>
</labels>
```

Général XML

Si la structure XML n'est pas supportée en interne par le logiciel, vous devrez définir le filtre XML et définir les règles d'extraction de données. Pour plus de renseignements, consultez l'article Comprendre les Filtres.

```
<?xml version="1.0" encoding="utf-8"?>
<asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
<asx:values>
<NICELABEL_JOB>
<TIMESTAMP>20130221100527.788134</TIMESTAMP>
<USER>PGRI</USER>
<IT_LABEL_DATA>
<LBL_NAME>goods_receipt.lbl</LBL_NAME>
<LBL_PRINTER>Production01</LBL_PRINTER>
<LBL_QUANTITY>1</LBL_QUANTITY>
<MAKTX>MASS ONE</MAKTX>
<MATNR>28345</MATNR>
<MEINS>KG</MEINS>
<WDATU>19.01.2012</WDATU>
<QUANTITY>1</QUANTITY>
<EXIDV>012345678901234560</EXIDV>
</IT_LABEL_DATA>
</NICELABEL_JOB>
</asx:values>
</asx:abap>
```

NiceLabel XML

Le traitement XML est intégré dans le logiciel.NiceLabel Vous ne devez pas configurer de filtres pour extraire les données, il suffit de lancer l'action intégrée Lancer le Fichier Commande. Pour plus d'informations concernant la structure XML, Voir l'article Fichier de Commande XML.

```
<nice_commands>
<label name="label1.lbl">

<session_print_job printer="CAB A3 203DPI" skip=0 job_name="job name 1" print_
to_file="filename 1">
<session quantity="10">
<variable name="variable name 1" >variable value 1</variable>
</session>
</session_print_job>

<print_job printer="Zebra R-402” quantity="10" skip=0 identical_copies=1 number_
of_sets=1 job_name="job name 2" print_to_file="filename 2">
<variable name="variable1" >1</variable>
<variable name="variable2" >2</variable>
<variable name="variable3" >3</variable>
</print_job>
</label>
</nice_commands>
```

Pour plus d'informations pratiques sur l'utilisation des données XML, consultez l'article Exemples.

Référence Et Résolution De Problèmes

Types De Fichiers De Commande

Caractéristiques Des Fichier De Commande

Les fichiers de commande contiennent les instructions pour le processus d'impression et sont exprimés avec les commandes NiceLabel. Les commandes sont exécutées une par une du début à la fin du fichier. Les fichiers supportent le formatage Unicode, vous pouvez donc y inclure des contenus multilingues.

Les fichiers de commande se présentent sous trois formes. Pour plus d'informations, consultez l'article Fichier de Commande JOB, Fichier de Commande CSV et Fichier de Commande XML.

Fichier De Commande CSV

Les commandes disponibles dans les fichiers de commande CSV sont un sous-ensemble des commandes NiceLabel. Vous pouvez utiliser les commandes: **LABEL**, **SET**, **PORT**, **PRINTER** et **PRINT**.

CSV signifie Comma Separated Values - Valeurs séparées par une virgule. C'est un fichier texte dans lequel les valeurs sont séparées par une virgule (`,`). Le fichier texte peut contenir une valeur Unicode (important pour les données multilingues). Chaque ligne dans le fichier de commande CSV contient les commandes pour une action d'impression d'étiquette.

La première ligne du fichier de commande CSV doit contenir les noms des commandes et des variables. L'ordre des noms de commandes n'est pas important, mais toutes les données dans le même flux d'impression doivent avoir la même structure. Les paires de variables `nom-valeur` sont extraites automatiquement et envoyées à l'étiquette de référence. Si la variable d'un nom du CSV n'existe pas dans l'étiquette, aucun message d'erreur n'est affiché.

Échantillon de fichier de Commande CSV

L'échantillon présente une vue structurelle des champs qui vous pouvez utiliser dans le fichier de commande CSV.

```
@Label,@Printer,@Quantity,@Skip,@IdenticalCopies,NumberOfSets,@Port,Product_ID,
Product_Name
label1.lbl, CAB A3 203 DPI, 100, , , , , 100A, Product 1
label2.lbl, Zebra R-402, 20, , , , , 200A, Product 2
```

Caractéristiques du fichier de Commande CSV

Les commandes dans la première digne de données doivent être exprimées avec le caractère (`@`). Les

champs sans @ au début sont des noms de variables, et ils seront extraits avec leurs valeurs comme paires `nom-valeur`.

- **@Label.** Spécifie le nom d'étiquette à utiliser. Il est conseillé d'inclure le chemin et nom du fichier. Assurez-vous que l'utilisateur du service peut accéder au fichier. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées. Un champ requis.
- **@Printer.** Spécifie l'imprimante à utiliser. Il remplace l'imprimante définie dans l'étiquette. Assurez-vous que l'utilisateur du service peut accéder à l'imprimante. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées. Champ optionnel.
- **@Quantity.** Spécifie le nombre d'étiquettes à imprimer. Valeurs possibles : valeur numérique, VARIABLE ou UNLIMITED. Pour plus d'informations, consultez l'article Impression d'étiquette. Un champ requis.
- **@Skip.** Spécifie le nombre d'étiquettes à sauter au début de la première page imprimée. Cette fonctionnalité est utile quand vous voulez réutiliser la feuille d'étiquettes partiellement imprimée. Champ optionnel.
- **@IdenticalCopies.** Spécifie le nombre de copies d'étiquettes à imprimer pour chaque étiquette unique. Cette fonctionnalité est utile à l'impression d'étiquettes contenant des données d'une base de données ou quand vous utilisez des compteurs et vous avez besoin de copies d'étiquettes. Champ optionnel.
- **@NumberOfSets.** spécifie le nombre de fois que le processus d'impression doit être répété. Chaque ensemble d'étiquettes définit l'occurrence du processus d'impression. Champ optionnel.
- **@Port.** Spécifie le nom du port pour l'imprimante. Vous pouvez remplacer le port par défaut spécifié dans le pilote d'imprimante. Vous pouvez également l'utiliser pour rediriger l'impression vers un fichier. Champ optionnel.
- **Autres noms de champs.** Tous les autres champs définissent les noms des variables de l'étiquette. Le contenu du champ sera sauvegardé dans la variable de même nom avec sa valeur.

Fichier De Commande JOB

Le fichier de commande JOB est un fichier texte contenant les commandes NiceLabel. Les commandes sont exécutées dans l'ordre, de haut en bas. Les commandes commencent habituellement par LABEL (pour ouvrir l'étiquette), ensuite SET (pour régler la valeur de variable) et finalement PRINT (pour imprimer l'étiquette). Pour plus d'informations concernant les commandes disponibles, consultez l'article Commandes Personnalisées.

Échantillon de fichier de Commande JOB

Ce fichier JOB va ouvrir `label2.lbl`, régler les variables et imprimer une étiquette. Comme aucune commande PRINTER n'est utilisée pour rediriger l'impression, l'étiquette sera imprimée en utilisant le nom d'imprimante défini dans l'étiquette.

```
LABEL "label2.lbl"
SET code="12345"
SET article="FUSILLI"
SET ean="383860026501"
SET poids="1,0 kg"
PRINT 1
```

Fichier De Commande XML

Les commandes disponibles dans les fichiers de commande XML sont un sous-ensemble des commandes NiceLabel. Vous pouvez utiliser les commandes suivantes : **LOGIN**, **LABEL**, **SET**, **PORT**, **PRINTER**, **SESSIONEND**, **SESSIONSTART** et **SESSIONPRINT**. La syntaxe change légèrement lors de l'utilisation dans un fichier XML.

L'élément de base dans le fichier de commande XML est `<Nice_Commands>`. L'élément suivant qui doit suivre est `<Label>`, et il spécifie l'étiquette à utiliser. Il y a deux méthodes pour lancer l'impression d'étiquettes : imprimer mes étiquettes normalement en utilisant l'élément `<Print_Job>`, ou imprimer les étiquettes en sessions, en utilisant l'élément `<Session_Print_Job>`. Vous pouvez également changer l'imprimante sur laquelle les étiquettes seront imprimées et paramétrer les valeurs de variables.

Fichier échantillon de Commande XML

L'échantillon présente une vue structurelle des éléments et leurs attributs tels qu'ils peuvent être utilisés dans le fichier XML.

```
<nice_commands>
<label name="label1.lbl">

<session_print_job printer="CAB A3 203DPI" skip=0 job_name="job name 1" print_to_
file="filename 1">
<session quantity="10">
<variable name="variable name 1" >variable value 1</variable>
</session>
</session_print_job>

<print_job printer="Zebra R-402” quantity="10" skip=0 identical_copies=1 number_of_
sets=1 job_name="job name 2" print_to_file="filename 2">
<variable name="variable1" >1</variable>
<variable name="variable2" >2</variable>
<variable name="variable3" >3</variable>
</print_job>
</label>
</nice_commands>
```

Caractéristiques des Commandes XML

Cette section contient la description de structure du fichier de commande XML. Il y a différents éléments qui contiennent des attributs. Certains attributs sont requis, d'autres ont optionnels. Certains attributs peuvent seulement occuper des valeurs pré-définies, pour d'autres vous pouvez définir des valeurs personnalisées.

- **<Nice_Commands>.** Ceci est un élément de base.
- **<Label>.** Spécifie le fichier d'étiquette à ouvrir. Si l'étiquette est déjà ouverte, elle ne sera pas réouverte. Le fichier d'étiquette doit être accessible depuis cet ordinateur. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées. Cet élément peut se présenter plusieurs fois dans le fichier de commande.
    - **Name.** Cet attribut contient le nom d'étiquette. Vous pouvez inclure le chemin vers le nom du fichier. Requis.
- **<Print_job>.** L'élément qui contient les données pour une tâche d'impression. Cet élément peut se présenter plusieurs fois dans le fichier de commande.

- **Printer.** Utilisez cet attribut pour substituer l'imprimante définie dans l'étiquette. L'imprimante doit être accessible depuis cet ordinateur. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées. Optionnel.
- **Quantity.** Utilisez cet attribut pour spécifier le nombre d'étiquettes à imprimer. Valeurs possibles : valeur numérique, VARIABLE ou UNLIMITED. Pour plus d'informations concernant les paramètres, consultez l'article Impression d'étiquette. Requis.
- **Skip.** Utilisez cet attribut pour spécifier le nombre d'étiquettes à sauter au début. Cet élément est utile si vous imprimez des feuilles d'étiquettes sur un imprimante laser, mais la feuille est déjà imprimée partiellement. plus d'informations, consultez l'article Impression d'étiquette. Optionnel.
- **Job_name.** Utilisez cet attribut pour spécifier le nom de votre tâche d'impression. Le nom spécifié est visible dans le spouleur d'impression. Pour plus d'informations, consultez l'article Définir le Nom de la Tâche d'impression. Optionnel.
- **Print_to_file.** Utilisez cet attribut pour spécifier le nom du fichier dans lequel vous voulez sauvegarder les commandes de l'imprimante. Pour plus d'informations, voir l'article Rediriger l'impression vers un fichier. Optionnel.
- **Identical_Copies**. utilisez cet attribut pour spécifier le nombre de copies que vous voulez pour chaque étiquette. Pour plus d'informations, consultez l'article Impression d'étiquette. Optionnel.

- **<Session_Print_Job>.** L'élément qui contient les données et les commandes pour une ou plusieurs sessions. L'élément peut contenir un ou plusieurs éléments `<Session>`. Il considère les règles d'impression de la session. Vous pouvez utiliser cet élément plusieurs fois dans le fichier de commande. Pour rechercher les attributs disponibles pour l'élément `<Print_Job>`. Ils sont tous valables, mais vous ne pouvez pas utiliser l'attribut de quantité. Voir la description de l'élément `<Session>` pour déterminer comment spécifier la quantité d'étiquettes dans la session d'impression.
- **<Session>.** L'élément qui contient les données pour une session. Quand vous imprimez en sessions, toutes les étiquettes sont codées dans une seule tâche d'impression et sont envoyées à l'imprimante comme une tâche.
  - **Quantity.** Utilisez cet attribut pour spécifier le nombre d'étiquettes à imprimer. Valeurs possibles : valeur numérique, chaîne de caractères VARIABLE ou chaîne UNLIMITED. Pour plus d'informations concernant les paramètres, consultez l'article Impression d'étiquette. Requis.
- **<Variable>.** L'élément qui paramètre la valeur des variables sur l'étiquette. Cet élément peut se présenter plusieurs fois dans le fichier de commande.
  - **Name.** L'attribut qui contient le nom de variable. Requis.

Schéma de Définition XML (XSD) pour le fichier de commande XML

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://tempuri.org/XMLSchema.xsd"
elementFormDefault="qualified" xmlns:mstns="http://tempuri.org/XMLSchema.xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="nice_commands">
<xs:complexType>
<xs:sequence>
<xs:element name="label" maxOccurs="unbounded" minOccurs="1">
```

```
<xs:complexType>
<xs:sequence>
<xs:element name="print_job" maxOccurs="unbounded" minOccurs="0">
<xs:complexType>
<xs:sequence>
<xs:element name="database" maxOccurs="unbounded" minOccurs="0">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="name" type="xs:string" use="required" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="table" maxOccurs="unbounded" minOccurs="0">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="name" type="xs:string" use="required" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="variable" maxOccurs="unbounded" minOccurs="0">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="name" type="xs:string" use="required" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="quantity" type="xs:string" use="required" />
<xs:attribute name="printer" type="xs:string" use="optional" />
<xs:attribute name="skip" type="xs:integer" use="optional" />
<xs:attribute name="identical_copies" type="xs:integer" use="optional" />
<xs:attribute name="number_of_sets" type="xs:integer" use="optional" />
<xs:attribute name="job_name" type="xs:string" use="optional" />
<xs:attribute name="print_to_file" type="xs:string" use="optional" />
<xs:attribute name="print_to_file_append" type="xs:boolean" use="optional" />
<xs:attribute name="clear_variable_values" type="xs:boolean" use="optional" />
</xs:complexType>
</xs:element>
<xs:element name="session_print_job" maxOccurs="unbounded" minOccurs="0">
<xs:complexType>
<xs:sequence>
<xs:element name="database" maxOccurs="unbounded" minOccurs="0">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="name" type="xs:string" use="required" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="table" maxOccurs="unbounded" minOccurs="0">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="name" type="xs:string" use="required" />
```

```
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="session" minOccurs="1" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="variable" minOccurs="0" maxOccurs="unbounded">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="name" type="xs:string" use="required" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="quantity" type="xs:string" use="required" />
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="printer" type="xs:string" use="optional" />
<xs:attribute name="skip" type="xs:integer" use="optional" />
<xs:attribute name="job_name" type="xs:string" use="optional" />
<xs:attribute name="print_to_file" type="xs:string" use="optional" />
<xs:attribute name="print_to_file_append" type="xs:boolean" use="optional" />
<xs:attribute name="clear_variable_values" type="xs:boolean" use="optional" />
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="name" type="xs:string" use="required" />
<xs:attribute name="close" type="xs:boolean" use="required" />
<xs:attribute name="clear_variable_values" type="xs:boolean" use="optional" />
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="quit" type="xs:boolean" use="required" />
</xs:complexType>
</xs:element>
</xs:schema>
```

Caractéristiques Oracle XML

Oracle a défini le format XML pour permettre la compréhension du contenu XML, son analyse et ensuite l'imprimer comme une étiquette. La Définition du Type de Document XML (DTD) définit les balises XML qui seront utilisées dans le fichier XML. Oracle va générer les fichiers XML selon cette DTD et le logiciel de partie ^{tierce} va traduire l'XML selon cette DTD.

Pour exécuter un tel fichier de commande, utilisez l'action Utiliser un fichier de Commande Oracle XML.

XML DTD

Ci-dessous, la DTD XML utilisée pour former la XML pour les formats XML synchrone et asynchrone, elle définit les éléments qui seront utilisés dans le fichier XML, une liste de leurs attributs et les éléments du niveau suivant.

```
<!ELEMENT labels (label)*>
<!ATTLIST labels _FORMAT CDATA #IMPLIED>
<!ATTLIST labels _JOBNAME CDATA #IMPLIED>
```

```
<!ATTLIST labels _QUANTITY CDATA #IMPLIED>
<!ATTLIST labels _PRINTERNAME CDATA #IMPLIED>
<!ELEMENT label (variable)*>
<!ATTLIST label _FORMAT CDATA #IMPLIED>
<!ATTLIST label _JOBNAME CDATA #IMPLIED>
<!ATTLIST label _QUANTITY CDATA #IMPLIED>
<!ATTLIST label _PRINTERNAME CDATA #IMPLIED>
<!ELEMENT variable (#PCDATA)>
<!ATTLIST variable name CDATA #IMPLIED>
```

Échantillon XML Oracle

Ceci est l'XML Oracle qui fournit les données pour une étiquette (il y a un seul élément `<label>`).

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE labels SYSTEM "label.dtd">
<labels _FORMAT ="Serial.lbl" _QUANTITY="1" _PRINTERNAME="" _JOBNAME="Serial">
<Label>
<variable name= "item">O Ring</variable>
<variable name= "revision">V1</variable>
<variable name= "lot">123</variable>
<variable name= "serial_number">12345</variable>
<variable name= "lot_status">123</variable>
<variable name= "serial_number_status">Active</variable>
<variable name= "organization">A1</variable>
</label>
</labels>
```

Lors de l'exécution de ce fichier XML Oracle, l'étiquette `serial.lbl` sera imprimée avec les valeurs suivantes.

Nom de variable.	Valeur de variable
item (article)	O Ring
révision	V1
lot	123
serial_number (numéro de série)	12345
lot_status (état du lot)	123
serial_number_status (état du numéro de série)	Active
organisation	A1

L'étiquette sera imprimée en `1` copie, avec le nom de tâche du spouleur `Série`. Le nom d'imprimante n'est pas spécifié dans le fichier XML, donc l'étiquette sera imprimée sur l'imprimante définie dans le modèle d'étiquette.

Caractéristiques SAP AII XML

NiceLabel Automation peut se présenter comme gestionnaire d'unité RFID, capable d'encoder et d'imprimer des étiquettes RFID. Pour plus d'information concernant les caractéristiques SAP AII XML, consultez le document **SAP Auto-ID Infrastructure Device Controller Interface** sur les pages Web SAP.

Pour exécuter un tel fichier de commande, utilisez l'action Utiliser un fichier de commande SAP AII XML.

Échantillon SAP AII XML

Ceci est l'XML AII SAP qui fournit les données pour une étiquette (il y a un seul élément `<label>`).

```
<?xml version="1.0" encoding="UTF-8"?>
<Command xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Command.xsd">
<WriteTagData readerID="DEVICE ID">
<Item>
<FieldList format="c:\SAP Demo\SAP label.lbl" jobName="Writer_Device20040929165746"
quantity="1">
<Field name="EPC">00037000657330</Field>
<Field name="EPC_TYPE">SGTIN-96</Field>
<Field name="EPC_URN">urn:autoid:tag:sgtin:3.5.0037000.065774.8</Field>
<Field name="PRODUCT">Product</Field>
<Field name="PRODUCT_DESCRIPTION">Product description</Field>
</FieldList>
<Item>
</WriteTagData>
</Command>
```

Lors de l'exécution de ce fichier SAP AI XML, l'étiquette `c:\SAP Demo\SAP label.lbl` sera imprimée avec les valeurs suivantes.

Nom de variable	Valeur de variable
EPC	00037000657330
EPC_TYPE	SGTIN-96
EPC	urn:autoid:tag:sgtin:3.5.0037000.065774.8
PRODUIT	Produit
PRODUCT_DESCRIPTION	Description

L'étiquette sera imprimée en `1` copie, avec le nom de tâche du spouleur `Writer_Device2004092916574`. Le nom d'imprimante n'est pas spécifié dans le fichier XML, donc l'étiquette sera imprimée sur l'imprimante définie dans le modèle d'étiquette.

Commandes Personnalisées

Utilisation De Commandes Personnalisées

Les commandes NiceLabel sont utilisées dans les fichiers de commande pour contrôler l'impression des étiquettes. NiceLabel Automation exécute la commande dans le fichier de commande, de haut en bas. Pour plus d'informations, voir l'article Référence et Résolution de Problèmes.

Vous pouvez utiliser la commande personnalisée spécifique, quand elle est disponible en tant qu'actions dans votre produit NiceLabel Automation.

NOTE: Par exemple, vous pouvez utiliser **SETPRINTPARAM**, si pouvez voir l'action **Set Print Parameter** (niveaux de produit Pro et Enterprise).

NiceLabelCaractéristiques de Commandes

COMMENTAIRE

```
;
```

Il est conseillé de documenter vos commandes durant le développement du fichier de commande. Cela vous aidera à décoder ce que le script vraiment fait, quand vous jetterez un coup d'œil sur le

script au bout d'un certain temps. Utilisez le point virgule (`;`) au début de la ligne. Tout ce qui suit le point-virgule sera considéré comme un commentaire et ne sera pas traité.

CLEARVARIABLEVALUES

```
CLEARVARIABLEVALUES
```

Cette commande remet les valeurs des variables à leur valeur par défaut.

CREATEFILE

```
CREATEFILE <file name> [, <contents>]
```

Cette commande va créer un fichier texte. Vous pouvez l'utiliser pour signaler à une partie tierce que le processus d'impression a commencé ou s'est terminé, selon l'endroit où vous placez la commande. Utilisez la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

DELETEFILE

```
DELETEFILE <nom de fichier>
```

Efface le fichier spécifié. Utilisez la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

IGNOREERROR

```
IGNOREERROR - ignorer erreur
```

Spécifie que l'erreur survenant dans le fichier JOB ne terminera pas le processus d'impression, si l'erreur suivante se produit :

- Nom de variable incorrect utilisé
- Une valeur incorrecte est envoyée à la variable
- L'étiquette n'existe pas / n'est pas accessible
- L'imprimante n'existe pas / n'est pas accessible

LABEL.

```
LABEL label_name [, printerName] (ÉTIQUETTE nom d'étiquette, nom d'imprimante)
```

La commande ouvre l'étiquette à imprimer. Si l'étiquette est déjà imprimée, elle ne sera pas réouverte. Vous pouvez inclure le nom du chemin. Mettez le nom de l'étiquette entre guillemets, si le nom ou le chemin contient des espaces. Utilisez la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

L'option `printer_name` spécifie l'imprimante pour laquelle l'étiquette sera ouverte. Utilisez ce paramètre si vous voulez remplacer le nom d'imprimante qui est sauvegardé dans le modèle d'étiquette. Si le pilote pour le nom d'imprimante fourni n'est pas installé ou pas disponible, la commande signalera une erreur.

MESSAGEBOX

```
MESSAGEBOX <message> [,<caption>]
```

Sauvegarde le `message` personnalisé dans le journal du déclencheur. Si le message des caractères d'espacement ou des virgules, vous devez placer le texte entre guillemets (").

PORT

```
PORT <file name> [, APPEND]
```

Cette commande remplace le port défini dans le pilote d'imprimante et redirige l'impression vers un fichier. Si le nom de chemin ou de fichier contient des espaces, mettez la valeur entre guillemets ("). Utilisez la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

Le paramètre `APPEND` est optionnel. Par défaut, le fichier sera remplacé. Utilisez ce paramètre pour ajouter des données à un fichier existant.

Quand vous utilisez une commande PORT dans le fichier JOB, il restera valable jusqu'à la commande PORT suivante, ou jusqu'à la fin du fichier (ce qui arrive en premier). Si vous utilisez une commande PRINTER après que la commande PORT a été exécutée, le paramètre PORT substituera le port défini pour l'imprimante sélectionnée. Si vous désirez utiliser le port actuel qui est défini pour l'imprimante sélectionnée, vous devez utiliser une autre commande PORT avec une valeur vide, telle que `PORT = ""`.

PRINT

```
PRINT quantity [, skip [, identical label copies [, number of label sets]]] -
Imprimer, quantité, saut, copies identiques, nombre de lots d'étiquettes.
```

Cette commande lance le processus d'impression.

- **Quantity.** Spécifie le nombre d'étiquettes à imprimer.
  - **<nombre>** Le nombre d'étiquettes spécifié sera imprimé.
  - **VARIABLE.** Spécifie qu'une variable d'étiquette est définie comme *quantité variable* et contiendra le nombre d'étiquettes à imprimer. L'étiquette va déterminer le nombre d'étiquettes à imprimer.
  - **UNLIMITED.** Si vous utilisez une base de données pour acquérir les valeurs des objets, l'impression illimité imprimera autant d'étiquettes qu'il y a de données dans la base de données. Si vous n'utilisez pas de base de données, le nombre maximal d'étiquettes supporté par l'imprimante thermique sera imprimé.
- **Skip.** Spécifie le nombre d'étiquettes que vous voulez sauter sur la première page. Le paramètre est utilisé pour l'impressiond'étiquettes sur des feuilles de papier. Quand la partie de la page a déjà été utilisée, vous pouvez réutiliser la même feuille en déplaçant le point de départ vers la première étiquette.
- **Copies Identiques d'étiquettes** Spécifie le nombre d'étiquettes identiques doivent être imprimées.
- **Nombre d'ensembles d'étiquettes.** Spécifie le nombre de fois que le processus d'impression complet doit être répété.

NOTE: Assurez-vous que les valeurs de quantité sont introduites en valeurs numériques, pas des chaînes de caractères. Ne placez pas les valeurs entre guillemets.

PRINTER

```
PRINTER <nom d'imprimante>
```

Cette commande remplace l'imprimante définie dans le fichier d'étiquette. Si le nom d'imprimante contient des caractères d'espacement, vous devez le mettre entre guillemets.

Utilisez le nom d'imprimante tel qu'il est affiché dans la ligne d'état de l'application de développement de l'étiquette. Les noms des imprimantes système sont généralement les mêmes que les noms d'imprimantes dans Imprimantes et Fax du Panneau de Contrôle, mais pas toujours. Quand vous utilisez des imprimantes réseau, vous pouvez voir le nom affiché sous la syntaxe `\\server\share`.

PRINTJOBNAME

```
PRINTJOBNAME - imprimer nom tâche
```

Cette commande spécifie le nom de la tâche d'impression que vous verrez dns le Spouleur Windows. Si le nom contient des caractères d'espacement ou virgules, vous devez le mettre entre guillemets (").

SESSIONEND

```
SESSIONEND - fin de session
```

Cette commande ferme le flux d'impression. Voir aussi **SESSIONSTART**.

SESSIONPRINT

```
SESSIONPRINT quantity [, skip] - impression de session, quantité, saut
```

Cette commande imprime les étiquettes actuellement référencées et l'ajoute à la session d'impression actuellement ouverte. Vous pouvez utiliser multiples commandes SESSIONPRINT l'une après l'autre et joindre les étiquettes référencées dans un flux d'impression unique. Le flux ne se fermera pas avant que vous le fermiez par la commande SESSIONEND. La signification des paramètres quantity (quantité) et skip (saut) est la même qu'avec NiceCommand PRINT. Voir aussi **SESSIONSTART**.

- **Quantity.** Spécifie le nombre d'étiquettes à imprimer.
- **Skip.** Spécifie le nombre d'étiquettes que vous voulez sauter sur la première page. Le paramètre est utilisé pour l'impressiond'étiquettes sur des feuilles de papier. Quand la partie de la page a déjà été utilisée, vous pouvez réutiliser la même feuille en déplaçant le point de départ vers la première étiquette.

SESSIONSTART

```
SESSIONSTART - début de session
```

Cette commande initie le type d'impression session-print.

Les trois commandes d'impression de session (**SESSIONSTART**, **SESSIONPRINT**, et **SESSIONEND**) sont utilisées ensemble. Quand vous utilisez la commande PRINT, toutes les données d'étiquettes sont envoyées à l'imprimante dans une tâche d'impression différente. Si vous voulez joindre les

données d'étiquettes dans un flux d'impression, vous devez utiliser les commandes d'impression de session. Vous devez commencer par la commande SESSIONSTART, suivie par un nombre quelconque de commandes SESSIONPRINT et terminer par la commande SESSIONEND.

Utilisez ces commandes pour optimiser le processus d'impression des étiquettes. L'impression d'étiquettes provenant d'un tâche d'impression est plus rapide que l'impression d'étiquettes de tâches d'impressions différentes.

Vous devez respecter certaines règles pour éviter de rompre la session d'impression.

- Vous ne pouvez pas changer l'étiquette pendant une session
- Vous ne pouvez pas changer d'imprimante pendant une session
- Vous devez paramétrer des valeurs pour toutes les variables de l'étiquette durant la session, même si certaines valeurs ont une valeur vide

SET

```
SET <nom>=<valeur> [pas> [,<nombre ou répétitions>]]
```

Cette commande assigne le `nom` de variable avec la `valeur`. La variable doit être définie sur l'étiquette, sinon une erreur sera signalée. Une erreur surviendra si la variable n'est pas sur l'étiquette. Le `Pas` et `nombre de répétitions` sont des paramètres pour les variables de comptage. Ces paramètres spécifient l'incrémentation du compteur et le nombre d'étiquettes avant que le compteur change de valeur.

Si la valeur contient des caractères d'espacement ou virgule, vous devez les mettre entre guillemets (`"`). Voir aussi **TEXTQUALIFIER**.

Si vous voulez assigner une valeur en plusieurs lignes, utilisez `\r\n` pour encoder un caractère de passage à la ligne. `\r` est remplacé par CR (Retour Charriot) et `\n` est remplacé par LF (Nouvelle Ligne).

Soyez prudent quand vous paramétrez les valeurs de variables procurant des données pour les images sur les étiquettes, comme le caractère barre oblique inverse qui peut être remplacé par un autre caractère.

> **EXEMPLE:** Si vous assignez la valeur "c:\Mes Images\raw.jpg" à la variable, le "\r" sera remplacé par le caractère CR.

SETPRINTPARAM

```
SETPRINTPARAM <nom paramètre> = <valeur>
```

Cette commande vous permet d'affiner les paramètres de l'imprimante avant d'imprimer. Les paramètres supportés pour les réglages d'imprimante (`paramname`) sont :

- **PAPERBIN.** Spécifie le réservoir contenant le média d'étiquette. Si l'imprimante est équipée de plus d'un bac à papier / bac à étiquettes, vous pouvez contrôler lequel sera utilisé pour l'impression. Le nom du bac à papier doit être acquis du pilote d'imprimante.
- **PRINTSPEED.** Spécifie la vitesse d'impression. Les valeurs acceptables varient d'une imprimante à l'autre. Consultez le manuel de l'imprimante pour la plage de valeurs exacte.

- **PRINTDARKNESS.** Spécifie l'obscurité / le contraste d'impression. Les valeurs acceptables varient d'une imprimante à l'autre. Consultez le manuel de l'imprimante pour la plage de valeurs exacte.
- **PRINTOFFSETX.** Spécifie la marge de gauche pour tous les objets d'impression. La valeur du paramètre doit être numérique, positive ou négative, en pixels.
- **PRINTOFFSETY.** Spécifie la marge supérieure pour tous les objets d'impression. La valeur du paramètre doit être numérique, positive ou négative, en pixels.
- **PRINTERSETTINGS.** Spécifie Le paramétrage personnalisé à appliquer à la tâche d'impression. Le paramètre requiert le DEVMODE entier pour l'imprimante ciblée, fourni dans une chaîne de caractères codée en Base64. Le DEVMODE contient tous les paramètres du pilote d'imprimante ensemble (vitesse, obscurité, décalage et autre). Pour plus d'informations, voir l'article Comprendre les paramètres d'imprimante et DEVMODE.

  NOTE: La chaîne de caractères codé en Base64 doit être fournie entre guillemets (").

TEXTQUALIFIER

```
TEXTQUALIFIER <caractère>
```

Text-qualifier (délimiteur de texte) est le caractère qui incorpore la valeur de donnée qui est assignée à une variable. Si la valeur comprend des caractères d'espacement, ils doivent être inclus dans des délimiteurs de texte. Le délimiteur de texte par défaut sont les guillemets ("). Comme les guillemets sont utilisés comme raccourci pour les unités de mesure en pouces, parfois il est difficile de passer les données avec le signe pouce dans les fichiers JOB. Vous pouvez utiliser des guillemets doubles ou utiliser TEXTQUALIFIER.

Exemple

```
TEXTQUALIFIER %
SET Variable = %EPAK 12"X10 7/32"%
```

Accès Aux Ressources Réseau Partagées

Cet article définit les recommandations à suivre quand vous utilisez des ressources de réseau partagées.

- **Privilèges utilisateur pour le mode de service.** Le composant d'exécution de NiceLabel Automation s'exécute en mode de service sous le compte utilisateur qui hérite les privilèges de ce compte. Pour que NiceLabel Automation puisse ouvrir les fichiers d'étiquettes et utiliser les pilotes d'imprimante, le compte utilisateur associé doit disposer des mêmes privilèges. Pour plus d'informations, consultez l'article Fonctionnement en Mode de Service
  .
- **notation UNC pour les partages réseau.** Quand vous accéder à un fichier sur un disque réseau, assurez-vous d'utiliser la syntaxe UNC (Convention Universelle de Noms) et pas les lettres des disques mappés. UNC est une convention de noms pour spécifier les disques réseau. NiceLabel Automation essaiera de remplacer la syntaxe de lettre du disque par la syntaxe UNC automatiquement.

EXEMPLE: Si le fichier est accessible comme `G:\Labels\label.lbl`, référenciez le en notation UNC comme `\\server\share\Labels\label.lbl` (où le disque G: est mappé à `\\server\share`).

- **Notation pour accéder au fichiers dans Control Center.** Quand vous ouvrez le fichier du le Stockage de Documents dans le Centre de Contrôle, vous pouvez utiliser la notation HTTP comme `http://servername:8080/label.lbl`, ou la notation WebDAV comme `\\servername@8080\DavWWWRoot\label.lbl`.

  Notes additionnelles :

  - Le compte utilisateur utilisé pour exécuter le service NiceLabel Automation sera utilisé pour récupérer les fichiers du Stockage de Documents. Cet utilisateur doit être paramétré dans la Configuration du Centre de Contrôle pour pouvoir accéder aux fichiers dans le Stockage de Documents.
  - L'accès WebDAV ne peut être utilisé qu'avec l'authentification d'utilisateur Windows dans le Centre de Contrôle.

  NOTE: Le Stockage de Documents est disponible avec les produits **NiceLabel Control Center Pro** et **NiceLabel Control Center Enterprise**.

- **Disponibilité de pilotes d'imprimantes.** Pour imprimer des étiquettes sur une imprimante réseau partagée, vous devez rendre le pilote disponible au serveur sur lequel NiceLabel Automation est installé. Vous pouvez installer me pilote d'imprimante manuellement et changer son port pour correspondre au socket d'imprimante (adresse IP et numéro de port), ou vous pouvez vous raccorder à l'imprimante par un serveur d'imprimante Dans ce cas, le compte utilisateur de service doit avoir accès au pilote d'imprimante.

Accéder Aux Bases De Données

Chaque fois que NiceLabel Automation doit accéder aux données d'une base de données, vous devez vous assurer que le pilote de base de donnée requis est installé dans le système Windows. Les pilotes de bases de données sont fournis par la société qui a développé le logiciel de la base de données. Le pilote que vous installez doit correspondre au nombre de bits de votre système Windows.

Windows 32-bits

Si vous avez Windows 32-bits, vous ne pouvez installer que des pilotes de base de données 32-bits. Le même pilote de base de données sera utilisé pour configurer le déclencheur dans le Automation Builder et effectuer l'exécution du déclencheur dans le Service NiceLabel Automation. Tous les composants NiceLabel Automation sont exécutés comme des applications 32-bits.

Windows 64-bits

Si vous avez Windows 64-bits, vous ne pouvez installer que des pilotes de base de données 32-bits ou 64-bits. Les applications exécutées en 64-bits utiliseront les pilotes de base de données 64-bits. Les applications exécutées en 32-bits utiliseront les pilotes de base de données 32-bits.

- **NiceLabel AutomationService.** Par défaut, le Service est exécuté en processus 64-bits, et nécessite donc une version 64-bit du pilote de base de données.

- **NiceLabel Automation Builder.** Il s'exécute toujours comme une application 32-bit. Comme tel, il doit utiliser les pilotes de base de donnée 32-bit. Donc, quand vous lancez un aperçu du déclencheur dans le Automation Builder, le déclencheur utilisera les pilotes de base de données 32-bit. Si vous déployez le même déclencheur dans le Automation Manager et l'exécutez en temps réel, le journal peut afficher un message d'erreur concernant le pilote de contenu de la base de données manquant. Cela arrivera si vous n'avez pas de pilote de base de données 64-bits installé sur le même système. Le Service s'exécute comme un processus 64-bits et requiert un pilote de base de données 64-bits.

AVERTISSEMENT Vous devez avoir les pilotes de base de données 32-bit et 64-bit installés simultanément si vous développez la configuration (dans Automation Builder) et la déployez (dans Automation Manager) sur la même machine 64-bit.

La seule exception à cette règle est d'avoir la connectivité aux sources de données Microsoft Excel et Microsoft Access. Quand les pilotes de base de données 64-bit ne sont pas disponnibles, le **Service** déchargera la tâche de connexion à la base de données vers le **Service Proxy**. Comme le processus du Service Proxy est en 32-bit, il utilisera les mêmes pilotes de base de données que ceux que vous avez utilisé dans le Automation Builder, et la connexion fonctionnera.

Changer Les Paramètres Par Défaut D'Impressions Multiples

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Chaque produit NiceLabel Automation peut tirer avantage des multiples-cœurs dans le processeur. Chaque cœur sera utilisé pour exécuter un processus d'impression. La moitié des cœurs est utilisée pour exécuter des tâches simultanées *normales* et l'autre moitié pour le traitement des tâches simultanées *session-impression*.

NOTE: En circonstances normales, vous ne devez jamais changer les paramètres par défaut. Assurez-vous que vous savez ce que vous faites quand vous changez ces paramètres par défaut.

Pour changer le nombre de tâches d'impression simultanées, effectuez les opérations suivantes :

1. Ouvrez le fichier `product.config` dans un éditeur de texte.
   Le fichier est ici :

   ```
   c:\ProgramData\EuroPlus\NiceLabel Automation\system.net\product.config
   ```

2. Changer les valeurs des éléments **MaxConcurrentPrintProcesses et** MaxConcurrentSessionPrintProcesses.

   ```
   <Configuration>
   <IntegrationService>
   <MaxConcurrentPrintProcesses>1</MaxConcurrentPrintProcesses>
   <MaxConcurrentSessionPrintProcesses>1</MaxConcurrentSessionPrintProcesses>
   ```

```
</IntegrationService>
</Configuration>
```

3. Sauvegardez le fichier. NiceLabel Automation mettra automatiquement à jour le service avec le nouveau nombre de tâches d'impression.

Impression en Session

L'impression en session s'active quand vous imprimez la même étiquette sur la même imprimante et imprimez beaucoup d'étiquettes. Toutes les étiquettes seront envoyées à l'imprimante dans une tâche d'impression. De l'autre côté on a l'impression sans-session où chaque étiquette est envoyée comme une tâche d'impression séparée. Du point de vue performances, l'impression en session est un meilleur choix. NiceLabel Automation détermine automatiquement le mode d'impression selon la configuration du déclencheur.

Compatibilité Avec Les Produits NiceWatch

NiceLabel Automation peut charger les configurations qui ont été définies dans un des produits NiceWatch. Dans la majorité des cas vous pouvez exécuter la configuration NiceWatch dans NiceLabel Automation sans aucune modification.

Les produits NiceLabel Automation Utilisent les nouveaux moteurs d'impression sur base .NET, optimalisés en performances et faible encombrement mémoire. Le nouveau moteur d'impression ne supporte pas toutes les options de création des étiquettes disponible dans le concepteur d'étiquette. Chaque nouvelle version de NiceLabel Automation diminue le fossé mais vous pouvez rencontrer des éléments non-disponibles.

Résoudre les problèmes d'incompatibilité

NiceLabel Automation vous préviendra si vous essayez d'imprimer des étiquettes existantes qui contiennent des fonctionnalités de conception qui ne sont pas disponibles dans le nouveau moteur d'impression.

S'il y a des incompatibilités entre les fichiers de configuration ou modèles d'étiquettes NiceWatch, vous serez notifié de :

- **Problèmes de configuration du déclencheur.** A l'ouverture de la configuration de NiceWatch (fichier .MIS), NiceLabel Automation la compare aux éléments supportés. Tous les éléments du produit NiceWatch ne sont pas disponibles dans NiceLabel Automation. Certains ne sont pas disponibles du tout, d'autres sont configurés différemment. Si le fichier MIS contient des éléments non-supportés, vous verrez une liste de ces éléments et ils seront enlevés de la configuration.

  Dans ce cas, vous devez ouvrir le fichier .MIS file dans Automation Builder et résoudre les problèmes d'incompatibilité. Vous devrez utiliser la fonctionnalité de NiceLabel Automation pour recréer la configuration.

- **Problèmes de modèles d'étiquettes.** Si vos modèles existants d'étiquettes contiennent des fonctionnalités pas supportées par le moteur d'impression fourni par le NiceLabel Automation, vous verrez des messages d'erreur dans l'onglet Journal. Cette information est visible dans Automation Builder (durant la création des déclencheurs) ou dans Automation Manager (quand vous exécutez les déclencheurs).

Dans ce cas, vous devez ouvrir le fichier d'étiquette dans le concepteur d'étiquettes et enlever les éléments non-supportés de l'étiquette.

NOTE: Pour plus d'informations concernant les incompatibilités avec NiceWatch et le concepteur d'étiquettes, consultez la Base de connaissances - article KB251.

Ouvrir la configuration NiceWatch pour l'éditer

Vous pouvez ouvrir la configuration NiceWatch existante (fichier .MIS) dans Automation Builder et l'éditerAutomation Builder. La configuration peut seulement être sauvegardée en format .MISX.

pour éditer la configuration NiceWatch, effectuez les opérations suivantes :

1. Démarrer Automation Builder.
2. Sélectionnez **Fichier>Ouvrir fichier NiceWatch**.
3. Dans l'interface de dialogue ouvert, naviguez vers le fichier de configuration NiceWatch (fichier .MIS).
4. Cliquez **OK**.
5. Si la configuration contient des fonctionnalités non-supportées, une liste d'éléments non-supportés sera affichée. Ils seront enlevés de la configuration.

Ouvrir la configuration NiceWatch pour l'exécuter

Vous pouvez ouvrir la configuration NiceWatch (fichier .MIS) dans Automation Manager sans conversion pour le format de fichier NiceLabel Automation (fichier .MISX). Vous pouvez commencer directement leur utilisation si les déclencheurs de NiceWatch sont compatibles avec NiceLabel Automation.

Pour éditer et déployer la configuration NiceWatch, effectuez les opérations suivantes :

1. Start Automation Manager.
2. Cliquez le bouton **+ Ajouter**.
3. Dans l'interface de dialogue **Ouvrir**, changez le type de fichier en **Configuration NiceWatch**.
4. Naviguez vers le fichier de configuration NiceWatch (fichier .MIS).
5. Cliquez **OK**.
6. Le déclencheur de la configuration sélectionnée sera affiché dans Automation Manager. Pour lancer le déclencheur, sélectionnez-le et cliquez le bouton **Start**.

NOTE: S'il y a un problème de compatibilité avec la configuration NiceWatch, vous devrez l'ouvrir dans Automation Builder et la reconfigurer.

Contrôler Le Service Avec Les Paramètres De Ligne De Commande

Ce chapitre fournit les informations pour démarrer et arrêter le Service d'Automation et comment contrôler quelles configurations sont chargées et quel déclencheurs sont actifs, depuis la ligne de commande.

NOTE: Assurez-vous de lancer la **Ligne de Commande** en mode élevé (avec les permissions administratives). Effectuez un clic-droit sur cmd.exe et sélectionnez **Exécuter en tant qu'Administrateur**.

Démarrer et arrêter les Services

Pour démarrer les deux services depuis la ligne de commande, utilisez les commandes suivantes :

```
net start NiceLabelAutomationProxyService
net start NiceLabelAutomationService
```

Si vous voulez ouvrir le fichier de configuration quand le Service est démarré, utilisez :

```
net start NiceLabelAutomationService [Configuration]
```

Par exemple :

```
net start NiceLabelAutomationService "c:\Project\configuration.MISX"
```

Pour arrêter les services, utilisez les commandes suivantes :

```
net stop NiceLabelAutomationProxyService
net stop NiceLabelAutomationService
```

Gérer les Configurations et Déclencheurs

Le service NiceLabel Automation peut contrôler les paramètres de la ligne de commande Automation Manager. La syntaxe générale d'utilisation de paramètres dans la ligne de commande est la suivante.

```
NiceLabelAutomationManager.exe COMMAND Configuration [NomDéclencheur] [/SHOWUI]
```

NOTE: Note : incluez le chemin complet vers le nom de configuration, n'utilisez pas le nom de fichier seul.

Pour ajouter (ADD) une configuration

La configuration fournie sera chargée dans le service. Aucun déclencheur ne sera démarré. Si vous incluez le paramètre `/SHOWUI`, Automation Manager UI sera démarré.

```
NiceLabelAutomationManager.exe ADD c:\Project\configuration.MISX /SHOWUI
```

Pour Recharger (RELOAD) la Configuration.

La configuration fournie sera rechargée dans le service. L'état d'exécution de tous les déclencheurs sera conservé. Recharger la configuration force la mise à jour de tous les fichiers en mémoire cache pour cette configuration. Pour plus d'informations, consultez l'article Mise en cache de Fichiers. Si vous incluez le paramètre `/SHOWUI`, Automation Manager UI sera démarré.

```
NiceLabelAutomationManager.exe RELOAD c:\Project\configuration.MISX /SHOWUI
```

Pour enlever (REMOVE) une configuration

La configuration fournie et tous ses déclencheurs seront déchargés du service.

```
NiceLabelAutomationManager.exe REMOVE c:\Project\configuration.MISX
```

Pour démarrer (START) un déclencheur

Le déclencheur référencé sera démarré dans la configuration déjà chargée.

```
NiceLabelAutomationManager.exe START c:\Project\configuration.MISX CSVTrigger
```

Pour arrêter (STOP) un déclencheur

Le déclencheur référencé sera arrêté dans la configuration déjà chargée.

```
NiceLabelAutomationManager.exe START c:\Project\configuration.MISX CSVTrigger
```

Codes d'états

Les codes d'états fournissent le feedback de l'exécution de la ligne de commande. Pour activer le retour des codes d'états, utilisez la syntaxe de ligne de commande suivante.

```
start /wait NiceLabelAutomationManager.exe COMMAND Configuration [TriggerName]
[/SHOWUI]
```

Les codes d'état sont capturés dans la variable système `errorlevel`. Pour voir le code d'état, exécutez la commande suivante.

```
echo %errorlevel%
```

Liste de codes d'états :

Code d'état	Description
0	Aucune erreur survenue
100	Nom du fichier de Configuration pas trouvé
101	La Configuration ne peut pas être chargée
200	Déclencheur pas trouvé
201	Le déclencheur ne peut pas démarrer

Introduction De Caractères Spéciaux (Codes De Contrôle)

Les caractères spéciaux ou codes de contrôle sont des caractères binaires qui ne sont pas représentés sur le clavier. Vous ne pouvez pas les taper comme des caractères normaux, ils doivent être codés par une syntaxe spéciale. Vous utiliserez ces caractères durant la communication avec des appareils à port série, recevant des données sur un port TCP/IP, ou pour travailler avec des fichiers binaires, tels que les fichiers d'impression.

Il existe deux méthodes pour introduire les caractères spéciaux :

- **Introduire les caractères spéciaux manuellement.** Vous pouvez introduire directement les caractères spéciaux en utilisant la syntaxe `<caractère_spécial_acronyme>`, telle que `<FF>` pour FormFeed, ou `<CR>` pour Retour Charriot, ou `<CR><LF>` pour une nouvelle ligne. pour plus d'informations, voir l'article Liste des Codes de Contrôle.

- **Insérer des caractères de la liste.** Les objets qui supportent les caractères spéciaux comme contenu affichent un bouton à flèche sur leur côté droit. Le bouton contient un raccourci vers tous les caractères spéciaux disponibles. Quand vous sélectionnez un caractère de la liste, il est ajouté au contenu. Pour plus d'informations, consultez l'article Utiliser des valeurs composées.

Liste Des Codes De Contrôle

Code ASCII	Abréviation	Description
1	SOH	Début de l'entête
2	STX	Début de Texte
3	ETX	Fin de Texte
4	EOT	Fin de Transmission
5	ENQ	Requête
6	ACK	Accusé de réception
7	BEL	Clochette
8	BS	Retour arrière
9	HT	Tabulation horizontale
10	LF	Nouvelle ligne
11	VT	Tabulation verticale
12	FF	Saut de page
13	CR	Retour charriot
14	SO	Extraction
15	SI	Introduction
16	DLE	Data Link Escape - Échappement Transmission
17	DC1	XON - Contrôle d'appareil 1
18	DC2	Contrôle d'appareil 2
19	DC3	XOFF - Contrôle d'appareil 3
20	DC4	Contrôle d'appareil 4
21	NAK	Accusé de réception Négatif
22	SYN	Veille Synchrone
23	ETB	Fin du bloc de Transmission
24	CAN	Annuler
25	EM	Fin de Média
26	SUB	Substituer
27	ESC	Échapper
28	FS	Séparateur de fichier
29	GS	Séparateur de Groupe
30	RS	Séparateur d'enregistrement
31	US	Séparateur d'unité
188	FNC1	Code de Fonction 1
189	FNC2	Code de Fonction 2
190	FNC3	Code de Fonction 3
191	FNC4	Code de Fonction 4

Mode Hors-ligne

La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Le mode hors-ligne Est un mode d'urgence qui s'active automatiquement quand le serveur de licence ne peut pas être contacté. Automation Manager affichera le message dans l'écran d'information et sauvegarder l'évènement dans le journal Windows Application Event. Quand NiceLabel Automation fonctionne en mode hors-ligne, il continuera à exécuter les déclencheurs durant 24 heures. Vous devrez rétablir une connexion au serveur de licence endéans 24 heures pou assurer une opération ininterrompue. Les informations d'activités d'impression seront conservés en mémoire cache locale et synchronisés avec le serveur dès que la connexion est rétablie.

NOTE: Le mode hors-ligne n'est que quand la licence NiceLabel Automation dans NiceLabel Control Center est activée.

Mode De Licence D'Imprimante

En fonction de la licence, votre produit NiceLabel Automation peut être limité au niveau du nombre d'imprimantes que vous pouvez utiliser simultanément. Dans ce cas, NiceLabel Automation conserve une trace du nombre et des noms des différentes imprimantes que vous avez utilisé pour imprimer. Quand vous dépassez le nombre d'imprimantes défini par votre licence, vous ne pouvez plus imprimer vers une nouvelle imprimante. L'erreur sera enregistrée dans le journal de la base de données.

Vous pouvez remettre la liste d'imprimantes à zéro en redémarrant le service NiceLabel Automation. Pour redémarrer le service, ouvrez Windows console de **Services** console, trouvez NiceLabel Automation Service et cliquez sur le bouton Redémarrer le Service.

AVERTISSEMENT Assurez-vous de redémarrer le service quand NiceLabel Automation est en veille. redémarrer le service dans un environnement chargé peut causer des défaillances de la production d'étiquettes. Quand le service est arrêté, toutes les listes d'attente et tampons sont vidés et les données intermédiaires effacées. Redémarrez le service avec prudence !

Fonctionnement En Mode De Service

NiceLabel Automation fonctionne comme un service Windows service et est conçu pour ne pas exiger d'intervention durant le traitement de données et l'exécution des actions. Le service est configuré pour démarrer quand le système d'exploitation est lancé et fonctionnera en arrière-plan tant que Windows fonctionne. NiceLabel Automation va mémoriser la liste de toutes les configurations chargées et déclencheurs actifs. Le dernier état connu est rétabli automatiquement quand le serveur redémarre.

Le service fonctionne avec les privilèges du compte utilisateur sélectionné durant l'installation. Le service hérite de toutes les permissions d'accès de ce compte utilisateur, y compris les ressources réseau partagées, telles que les disques réseau et pilotes d'imprimantes. Utilisez le compte d'un

utilisateur existant ayant des privilèges suffisant, ou mieux, créez un compte dédié pour NiceLabel Automation.

Vous pouvez gérer le service en lançant le Services depuis le panneau de contrôle Windows. Dans les systèmes d'exploitation modernes, vous pouvez également gérer le service dans l'onglet Services de Windows Gestion de Tâches. Vous utiliserez Services pour exécuter des tâches telles que :

- Démarrer et arrêter le service
- Changer le compte sous lequel le service se connecte

Conseils de Configuration du Compte Utilisateur pour Service

- Même si c'est possible, on déconseille vivement d'exécuter le service sous le Compte Local du Système. C'est un compte prédéfini Windows ayant des privilèges étendus sur l'ordinateur local, mais ne dispose généralement pas de privilèges d'accès au ressources du réseau. NiceLabel Automation requiert l'accès complet au dossier `%temp%` du compte, qui ne doit pas être disponible au Compte du Local du Système.
- Si vous créez un **nouveau compte utilisateur** pour le service NiceLabel Automation, assurez-vous de vous connecter à Windows en utilisant ce nouvel utilisateur au moins une fois. Cela assurera que le compte utilisateur est créé complètement. Par ex. quand vous vous connectez, le dossier temporaire `%temp%` sera créé.
- Désactivez la nécessité de changer le mot de passe de temps en temps pour ce compte utilisateur.
- Assurez-vous que le compte a les permissions de se **Connecter comme un service**.
- Exécutez le Service en mode x64 (64-bits).

Accéder aux Ressources

NiceLabel Automation hérite de tous les privilèges du compte utilisateur Windows sous lequel le service s'exécute. Le service exécute toutes les actions sous ce nom de compte. L'étiquette peut être ouverte si le compte a les permissions d'accès au fichier. L'étiquette peut être imprimée si le compte à accès au pilote d'imprimante.

Pour plus d'informations, consultez l'article Accès aux ressources réseau partagées.

Mode Service : 32-bit vs 64-bit

NiceLabel Automation peut fonctionner sur les systèmes natifs 32-bits (x86) et 64-bits (x64). Le mode d'exécution est déterminé automatiquement par le système d'exploitation Windows. NiceLabel Automation fonctionnera en mode 64-bits sur Windows 64-bits et fonctionnera en mode 32-bits sur Windows 32-bits.

- **Impression.** Il y a des bénéfices à fonctionner comme un processus 64-bits, tel que la communication directe avec le service de spouleur d'imprimante 64-bit sur Windows 64-bits. Ceci élimine les fameux problèmes avec SPLWOW64.EXE, qui est un 'intergiciel' pour permettre aux applications 32-bits d'utiliser le service de spouleur d'imprimante 64-bits.
- **Accès à la base de données.** Le fonctionnement des processus 64-bits NiceLabel Automation Service requiert une version 64-bits des pilotes de la base de données pour accéder aux données. Pour plus d'informations, voir l'article Accéder aux bases de données.

NOTE: Si vous ne disposez pas de pilotes de base de données 64-bits pour votre base de données, vous ne pouvez pas utiliser NiceLabel Automation en mode 64-bits. Vous devez l'installer sur un système 32-bits, ou le forcer en mode 32-bits.

Forcer le mode d'Opération x86 sur Windows x64

Vous pouvez avoir des raisons d'utiliser NiceLabel Automation en application 32-bits sur Windows 64-bits.

Pour forcer NiceLabel Automation en mode x86 sur Windows x64, effectuez les opérations suivantes :

- Sélectionnez Start ->Run.
- Tapez **regedit** et appuyez Enter
- Naviguez vers la clé

```
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\services\NiceLabelAutomationService
```

- Changez le nom de fichier en **NiceLabelAutomationService.x86.exe**, en conservant le chemin existant.
- Redémarrez le service NiceLabel Automation.

AVERTISSEMENT Il n'est pas recommandé de changer mode de service NiceLabel Automation. Si vous allez quand même le faire, assurez-vous d'effectuer des tests étendus du déclencheur avant le déploiement dans l'environnement de production.

Ordre De Recherche Pour Les Fichiers Requis

Quand NiceLabel Automation essaie de charger un fichier d'étiquette ou un fichier image spécifique il n'annulera pas le processus et ne rapporte pas d'erreur si le fichier n'est pas trouvé. Il essaiera de trouver le fichier requis dans les autres emplacements.

NiceLabel Automation essaiera de trouver le fichier dans l'ordre suivant :

1. Vérifie si le fichier existe à l'endroit spécifié par l'action.
2. Vérifie si le fichier existe dans le même dossier que le fichier de configuration (.MISX).
3. Vérifie si le fichier d'étiquette existe dans le dossier .\Labels (pour les fichiers graphiques il vérifie le dossier .\Graphics).
4. Vérifie si le fichier d'étiquette existe dans le dossier ..\Labels (pour les fichiers graphiques il vérifie le dossier ..\Graphics).
5. Vérifie si le fichier existe dans le dossier global Labels (Dossier Graphics pour les fichiers graphiques files) dans l'utilitaire de Configuration.

Si le fichier n'existe dans aucun de ces endroits, l'action échoue et une erreur est signalée.

Sécuriser L'accès À Vos Déclencheurs

Dans certains déploiements, vous voulez sécuriser l'accès à vos déclencheurs. NiceLabel Automation permet d'activer des mesures de sécurité pour permettre l'accès aux déclencheurs par les appareils de confiance du réseau. La configuration de sécurité dépend du type de déclencheur. Certains types de déclencheurs permettent la configuration de sécurité d'accès par leur conception. Pour tous les déclencheurs basés sur le protocole TCP/IP, vous pouvez définir tous les détails dans le pare-feu Windows.

Configuration du Pare-feu

Quand vous utilisez des déclencheurs sur base TCP/IP, tels que Déclencheur Serveur TCP/IP, assurez-vous de mettre aux applications externes de se connecter aux déclencheurs. Chaque déclencheur fonctionne dans le service NiceLabel Automation, dont l'accès est gouverné par le Pare-feu Windows. Un pare-feu fonctionne comme la serrure de votre porte d'entrée - il aide à maintenir les intrus à l'extérieur.

NOTE: Par défaut, le pare-feu Windows est configuré pour permettre toutes les connexions entrantes au service NiceLabel Automation. Ceci rend la configuration et l'essai des déclencheurs plus aisée, mais peut être susceptible à un accès non-autorisé.

Si le déploiement de NiceLabel Automation dans votre société est un sujet de régulations de sécurité strictes, vous devez mettre les règles de votre pare-feu à jour en conséquence.

Par exemple :

- Vous pouvez affiner le pare-feu pour accepter seulement le trafic entrant de sources bien connues.
- Vous pouvez autoriser les données entrantes seulement sur les ports prédéfinis.
- Vous pouvez permettre la connexion de certains utilisateurs.
- Vous pouvez définir sur quels interfaces vous allez accepter les connexions entrantes.

Pour effectuer des changements au pare-feu Windows, ouvrez la console de gestion **Pare-feu Windows avec Sécurité Avancée dans le** Panneau de **Contrôle-> Système et Sécurité -> Pare-feu Windows -> Paramètres Avancés**.

Permettre l'Accès Basé sur les Permissions d'Accès Fichier

Le déclencheur de fichier va exécuter le changement de l'évènement d'horodatage dans le ou les fichiers surveillés. Vous devez mettre les fichiers du déclencheur dans un dossier accessible pour le service NiceLabel Automation. Le compte utilisateur sur lequel le Service fonctionne doit pouvoir accéder aux fichiers. Simultanément, les permissions d'accès aux emplacements déterminent quel utilisateur et ou application peut sauvegarder le fichier du déclencheur. Vous devez paramétrer les permissions d'accès de façon à ce que seuls les utilisateurs autorisé peuvent sauvegarder les fichiers.

Permission d'accès basée sur l'adresse IP & le nom d'hôte.

Vous pouvez protéger l'accès au déclencheur du serveur TCP/IP avec deux listes d'adresses IP et noms d'hôtes.

- La première liste **'Permettre la connexion des hôtes suivants'** contient les adresses IP ou noms d'hôtes d'appareils qui peuvent envoyer des données au déclencheur. Quand un appareil a une adresse IP de cette liste, il est autorisé à envoyer des données au déclencheur.
- La seconde liste **'Interdire la connexion pour les hôtes suivants'** contient des adresses IP ou noms d'hôtes d'appareils qui ne sont pas autorisés à envoyer des données. Quand un appareil a une adresse IP de cette liste, il n'est pas autorisé à envoyer des données au déclencheur.

Conseils Et Astuces D'utilisation De Variables Dans Les Actions

Suivez les recommandations ci-dessous quand vous utilisez des variables pour les actions dans le NiceLabel Automation.

- **Mettez les variables entre crochets carrés.** Quand vous avez des variables contenant des espaces dans leur nom qui se réfèrent à des variables dans des actions, telles que Exécuter une instruction SQL ou Exécuter un Script, entourez les variables par des crochets carrés, comme `[Nom Produit]`. Vous utiliserez également les crochets carrés si les noms de variable sont des noms réservés, par ex. dans les Instructions SQL.
- **Placez un double-point devant le nom de variable.** Pour référer à une variable dans l'instruction Exécuter une instruction SQL ou dans un Déclencheur de Base de Données, vous devez placer un double-point (`:`) devant le nom de variable, tel que `[Product ID]`. L'analyseur SQL le comprendra comme 'valeur variable'.

```
SELECT * FROM MyTable WHERE ID = :[ProductID]
```

- **Convertir des valeurs en intègres pour le calcul.** Quand vous voulez exécuter un calcul numérique avec les variables, assurez-vous de convertir la variable en valeur intègre. Définir les variables en numérique limite seulement les caractères acceptés pour la valeur mais ne change pas le type de la variable. NiceLabel Automation traite toutes les variables de type chaîne de caractères. Dans VBScript vous utiliserez la fonction `CInt()`.
- **Paramètres par défaut / valeurs de démarrage des scripts.** Quand vous utilisez des variables dans l'action Exécuter un Script assurez-vous qu'elles ont une valeur par défaut, autrement l'essai du script peut échouer. Vous pouvez définir les valeurs par défaut dans les caractéristiques des variables, ou dans le script (et les enlever après avoir testé le script).

Mode De Traçage

Par défaut, NiceLabel Automation enregistre les évènements dans la base de données. Ceci inclut les informations de niveau supérieur, telles que le rapportage d'exécution des actions, le rapportage d'exécution de filtrage et le rapportage des mises à jour d'état des déclencheurs. Pour plus d'informations, consultez l'article Options de journalisation des évènements.

Toutefois, le rapportage par défaut n'enregistre pas les détails inférieurs des exécutions. Le mode de traçage doit être activé quand une résolution de problème en profondeur est requise pour l'exécution du code. Dans ce mode, NiceLabel Automation enregistre les détails de toutes les exécutions internes qui se déroulent durant le traitement du déclencheur. Le mode de traçage ne doit

être activé que pour la résolution de problèmes pour collecter les données et ensuite désactivé pour permettre le mode d'opération normal.

NOTE: Le mode de traçage va ralentir le traitement et ne devrait être utilisé que quand l'équipe de support NiceLabel vous le demande.

Activer le mode de traçage

Pour activer le mode de traçage, effectuez les opérations suivantes :

1. Naviguez vers le dossier NiceLabel Automation System.Net.

   ```
   c:\ProgramData\EuroPlus\NiceLabel Automation\system.net
   ```

2. Effectuez une copie de sauvegarde du fichier `product.config`.
3. Ouvrez `product.config` dans un éditeur de texte. Le fichier a une structure XML.
4. Ajoutez l'élément `Common/Diagnostics/Tracing/Enabled` et assignez-lui la valeur **True**.

   Le fichier devrait avoir le contenu suivant :

   ```
   <?xml version="1.0" encoding="utf-8"?>
   <Configuration>
   <Common>
   <Diagnostics>
   <Tracing>
   <Enabled>True</Enabled>
   </Tracing>
   </Diagnostics>
   </Common>
   ...
   </Configuration>
   ```

5. Quand vous sauvegardez le fichier, le Service NiceLabel Automation va appliquer les paramètres automatiquement.
6. Les fichiers de traçage (*.LOG) apparaitront dans le même dossier.
7. Démarrez Automation Manager pour confirmer que le mode de traçage est activé. Il affichera le texte **Le traçage a été activé** dans l'interface de notification au-dessus de la liste du déclencheur.

Comprendre Les Paramètres D'imprimante Et DEVMODE

NOTE: La structure de données DEVMODE fait partie de Structure GDI d'Impression API de Windows. Ce sont des informations techniques avancées, utilisées seulement pour des requêtes très spécifiques.

Chaque fois que vous imprimez l'étiquette avec le logiciel NiceLabel (ou un document dans une application Windows appropriée), l'application d'impression va lire les paramètres de l'imprimante définis dans le pilote d'imprimante et les appliquer à la tâche d'impression. La même étiquette peut

être imprimée sur différentes imprimantes en sélectionnant un pilote d'imprimante différent. Les paramètres d'imprimante d'une nouvelle imprimante s'appliquent chaque fois à l'étiquette.

L'impression d'un document de texte sur une autre imprimante laser produit généralement un résultat identique ou comparable. L'impression d'étiquettes sur l'une ou l'autre imprimante laser pourrait produire des résultats différents. Un même fichier d'étiquette peut demander des paramètres supplémentaires dans le pilote d'imprimante, tels que des réglages de marges, vitesse et température d'impression pour produire des résultats comparables. NiceLabel applique aussi les paramètres d'imprimante à chaque impression. Par défaut, les paramètres d'imprimante sont sauvegardés dans le fichier d'étiquette pour l'imprimante sélectionnée.

Définition de DEVMODE?

DEVMODE est une structure Windows qui contient les paramètres d'imprimante (informations d'initialisation et d'environnement de l'imprimante). Il est constitué de deux parties : public et privé. La partie publique contient les données communes à toutes les imprimantes. La partie privée contient les données spécifiques à une imprimante particulière. La partie privée peut avoir une longueur variable et contenir tous les paramètres particuliers pour un fabriquant.

- **Partie publique.** Cette partie encode les paramètres généraux qui sont exposés dans le modèle du pilote d'imprimante, tels que le nom d'imprimante, la version du pilote, la taille du papier, l'orientation, la couleur, duplex et similaires. La partie publique est la même pour tout pilote d'imprimante et ne supporte pas les spécificités des imprimantes d'étiquettes (imprimantes thermales, imprimante à jet d'encre industrielles, machines de gravure laser).
- **Partie privée.** Cette partie encode les paramètres qui ne sont pas disponibles dans la partie publique. Les pilotes d'imprimante NiceLabel utilisent cette partie pour sauvegarder les données spécifiques au modèle, tels la vitesse d'impression, les paramètres de chauffage, les décalages, le mode d'impression, le type de média, capteurs, couteaux, encodage graphique, support RFID et similaires. La structure de données dans la partie publique dépend du développeur de pilote et se présente sous la forme d'un flux de données binaires.

Changer le DEVMODE

La structure de données DEVMODE est stockée dans les registres Windows. Il y a deux copies de la structure : les paramètres d'imprimante par défaut et les paramètres d'imprimante spécifiques à l'utilisateur. DEVMODE (paramètres d'imprimante) peut être modifié en changeant les paramètres dans le pilote d'imprimante. Les deux premières options sont liées à Windows, la troisième option est disponible dans le logiciel NiceLabel.

- **Paramètres d'imprimante par défaut.** Ils sont définis dans **Caractéristiques d'imprimante>Onglet Avancés>Par.Défaut Imprimante**.
- **Paramètres utilisateur spécifiques.** Ils sont sauvegardés séparément pour chaque utilisateur dans les clés de registres utilisateurs HKEY_CURRENT_USER. Par défaut, les paramètres utilisateur spécifiques sont hérités des paramètres d'imprimante par défaut. Les paramètres utilisateur spécifiques sont définis dans **Caractéristiques Imprimante>Préférences**. Toutes les modifications apportées ici n'affecteront que l'utilisateur actuel.
- **Paramètres d'étiquette spécifiques.** Le développeur utilisant le logiciel NiceLabel NiceLabel peut choisir d'incorporer le DEVMODE dans l'étiquette elle-même. Ceci permet le déplacement des paramètres d'imprimante. Quand l'étiquette est copiée sur un autre ordinateur, les paramètres d'imprimante la suivent. Pour incorporer les paramètres dans l'étiquette, activez l'option **Utiliser les paramètres personnalisés sauvegardés dans l'étiquette** dans

*Fichier>Paramètres Étiquette>Onglet Imprimante*Designer Pro dans le . Vous pouvez modifier les paramètres d'imprimante dans l'étiquette en sélectionnant *Fichier>Paramètres Imprimante*.

Appliquer le DEVMODE personnalisé à l'impression

Dans NiceLabel Automation vous pouvez ouvrir un fichier d'étiquette et lui appliquer le DEVMODE personnalisé A l'impression de l'étiquette, le design de l'étiquette est pris du fichier .LBL et le DEVMODE applique le formatage spécifique lié à l'imprimante. Ceci vous permet une seule étiquette Maître. L'impression sera la même, peu importe l'imprimante utilisée, car les paramètres optimaux d'impression sont appliqués pour cette imprimante.

Pour appliquer le DEVMODE à l'étiquette, vous pouvez utiliser deux options :

1. Utiliser l'action Définir les Paramètres d'Impression, plus spécifiquement le paramètre **Paramètres d'imprimante**.
2. Le fichier de commande JOB, plus spécifiquement la commande **SETPRINTPARAM** avec le paramètre **PRINTERSETTINGS**. Pour plus d'informations, voir Commandes Personnalisées.

Utiliser Le Même Compte Utilisateur Pour Configurer Et Exécuter Les Déclencheurs

Le Service NiceLabel Automation s'exécute toujours sous les infos d'identification du compte utilisateur configuré pour le service. Toutefois, Automation Builder s'exécute toujours sous les infos d'utilisateur de l'utilisateur actuellement connecté. Les infos d'utilisateur du compte de service et du compte connecté peuvent être différents. Même si vous pouvez voir l'aperçu du déclencheur dans le Automation Builder sans problèmes, le Service peut rapporter un message d'erreur, causé par une différence d'infos utilisateurs. Si l'utilisateur actuellement connecté a la permission d'accéder aux dossiers et imprimantes, le comte d'utilisateur Service utilisé par Service ne doit pas l'avoir.

Vous pouvez tester l'exécution des déclencheurs dans Automation Builder en utilisant les mêmes infos d'utilisateur que celles du Service. Pour cela, utilisez Automation Builder sous le même compte utilisateur que celui défini pour le Service.

Effectuez les opérations suivantes pour utiliser Automation Builder sous un autre compte :

1. Appuyez et maintenez la touche **Shift**, ensuite **clic-droit** sur l'icône Automation Builder.
2. Sélectionnez **Utiliser sous un autre compte utilisateur**.
3. Introduisez les infos d'utilisateur pour le même utilisateur que celui défini dans NiceLabel Automation Service.
4. Cliquez **OK**.

Si vous désirez utiliser fréquemment Automation Builder avec les infos utilisateur de l'autre compte utilisateur, voir l'utilitaire de la ligne de commande Windows **RUNAS.** Utilisez les commutations `/user` pour spécifier le compte utilisateur et `/savecred` pour introduire le mot de passe qu'une seule fois, il sera ainsi mémorisé pour la fois suivante.

Exemples

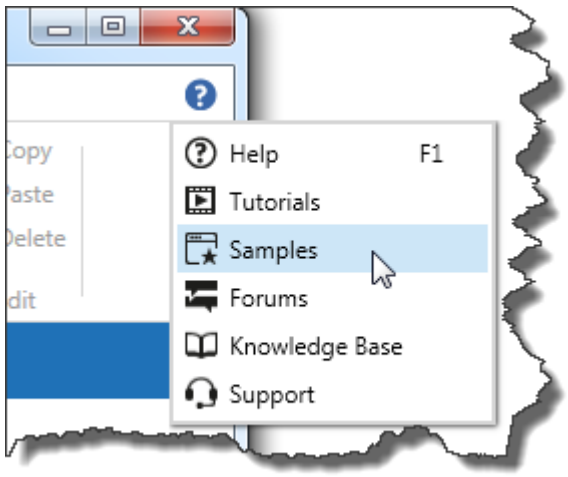
Exemples

NiceLabel Automation est livré avec des exemples qui décrivent les procédures de configuration pour les structures de données fréquemment utilisées et la configuration d'actions. Vous pouvez rapidement vous familiariser avec la configuration de filtres pour extraire les données des fichiers CSV (Comma Separated Values), de données de base exportées, de fichiers d'impression, de documents XML, de fichiers binaires, et autres.

Un raccourci vers le dossier d'échantillons est disponible dans Automation Builder.

Pour ouvrir le dossier d'échantillons dans Windows Explorer, effectuez les opérations suivantes :

1. Ouvrez Automation Builder.
2. Cliquez sur le point d'interrogation dans le coin supérieur droit.
3. Sélectionnez **Échantillons**.



4. Le dossier contenant les fichiers d'exemples s'ouvrira dans Windows Explorer.
5. Consultez le fichier **README.PDF** dans chaque dossier.

Les échantillons sont installés dans le dossier suivant, en fonction du système que vous utilisez.

- Si vous avez Windows XP

EXEMPLE: %ALLUSERSPROFILE%\Documents\NiceLabel Samples\Automation

qui se résoudra dans
C:\Documents and Settings\All Users\Documents\NiceLabel Samples\Automation

- Si vous avez Windows 7 / Windows Server 2008 et supérieur

EXEMPLE:

c:\Users\Public\Documents\NiceLabel Samples\Automation qui se résoudra dans **c:\Users\Public\Documents\NiceLabel Samples\Automation**

Support Technique

Support En-ligne

Vous trouverez les dernières versions, mises à jour, solutions de contournement des problèmes et les Questions Fréquemment Posées (FAQ) sur les pages du site web www.nicelabel.com.

Pour plus d'informations, veuillez vous référer à :

- Base de connaissances : http://kb.nicelabel.com
- Support NiceLabel : http://www.nicelabel.com/support
- Didacticiels NiceLabel : www.nicelabel.com/Learning-center/Tutorials
- Forums NiceLabel : forums.nicelabel.com

NOTE: Si vous avez un Contrat de Support (Service Maintenance Agreement - SMA), veuillez contacter le support Premium, comme spécifié dans le contrat.